Ildar Batyrshin
Miguel González Mendoza (Eds.)

LNAI 7630

Advances in Computational Intelligence

11th Mexican International Conference
on Artificial Intelligence, MICAI 2012
San Luis Potosí, Mexico, October/November 2012
Revised Selected Papers, Part II

2 Part II

MICAI
2012

Springer

Lecture Notes in Artificial Intelligence 7630

Subseries of Lecture Notes in Computer Science

Ildar Batyrshin
Miguel González Mendoza (Eds.)

Advances in Computational Intelligence

11th Mexican International Conference
on Artificial Intelligence, MICAI 2012
San Luis Potosí, Mexico, October 27 – November 4, 2012
Revised Selected Papers, Part II

Springer

Volume Editors

Ildar Batyrshin
Mexican Petroleum Institute
Eje Central Lazaro Cardenas Norte, 152
Col. San Bartolo Atepehuacan
México D.F., CP 07730, Mexico
E-mail: batyr1@gmail.com

Miguel González Mendoza
Tecnológico de Monterrey
Campus Estado de México
Carretera Lago de Guadalupe Km 3.5
Atizapán de Zaragoza
Estado de México, CP 52926, Mexico
E-mail: mgonza@itesm.mx

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-37797-6 e-ISBN 978-3-642-37798-3
DOI 10.1007/978-3-642-37798-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013935145

CR Subject Classification (1998): I.2.1, I.2.3-11, I.4, I.5.1-4, J.3, H.4.1-3, F.2.2, H.3.3-5, H.5.3, F.1.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

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 Mexican International Conference on Artificial Intelligence (MICAI) is a yearly international conference series that has been organized by the Mexican Society of Artificial Intelligence (SMIA) since 2000. MICAI is a major international AI forum and the main event in the academic life of the country's growing AI community.

MICAI conferences publish high-quality papers in all areas of AI and its applications. The proceedings of the previous MICAI events have been published by Springer in its *Lecture Notes in Artificial Intelligence* (LNAI) series, vol. 1793, 2313, 2972, 3789, 4293, 4827, 5317, 5845, 6437, 6438, 7094, and 7095. Since its foundation in 2000, the conference has been growing in popularity and improving in quality.

The proceedings of MICAI 2012 are published in two volumes. The first volume, *Advances in Artificial Intelligence*, contains 40 papers structured into five sections:

- Machine Learning and Pattern Recognition
- Computer Vision and Image Processing
- Robotics
- Knowledge Representation, Reasoning, and Scheduling
- Medical Applications of Artificial Intelligence

The second volume, *Advances in Computational Intelligence*, contains 40 papers structured into four sections:

- Natural Language Processing
- Evolutionary and Nature-Inspired Metaheuristic Algorithms
- Neural Networks and Hybrid Intelligent Systems
- Fuzzy Systems and Probabilistic Models in Decision Making

Both books will be of interest for researchers in all fields of AI, students specializing in related topics, and for the general public interested in recent developments in AI.

The conference received for evaluation 224 submissions by 461 authors from 28 countries: Algeria, Austria, Belgium, Brazil, Canada, China, Colombia, Cuba, Czech Republic, France, Hungary, India, Iran, Israel, Japan, Luxembourg, Mexico, The Netherlands, New Zealand, Pakistan, Paraguay, Peru, Poland, Russia, Spain, Tunisia, UK, and USA. Of these submissions, 77 papers were selected for publication in these two volumes after a peer-reviewing process carried out by the international Program Committee. The acceptance rate was 34.3%.

MICAI 2012 was honored by the presence of such renowned experts as Ulises Cortés of the Universitat Politècnica de Catalunya, Spain; Joydeep Ghosh of the University of Texas, USA; Jixin Ma of Greenwich College, UK; Roy A. Maxion of

Carnegie Mellon University, USA; Grigori Sidorov of the Instituto Politécnico Nacional, Mexico; and Ian Witten of the University of Waikato, New Zealand, who gave excellent keynote lectures. The technical program of the conference also featured nine tutorials presented by Ulises Cortes, Alexander Gelbukh, Jean Bernard Hayet, Sergio Ledezma, Jixin Ma, Roy A. Maxion, Horacio Rostro González, Grigori Sidorov, and Ian Witten. Two workshops were held jointly with the conference: the 5th Workshop on Intelligent Learning Environments and the 5th Workshop on Hybrid Intelligent Systems.

In particular, in addition to regular papers, the volumes contain three invited papers by keynote speakers and their collaborators:

- "Empirical Study of Machine Learning-Based Approach for Opinion Mining in Tweets," by Grigori Sidorov, Sabino Miranda-Jiménez, Francisco Viveros-Jiménez, Alexander Gelbukh, Noé Castro-Sánchez, Francisco Velásquez, Ismael Díaz-Rangel, Sergio Suárez-Guerra, Alejandro Treviño, and Juan Gordon
- "AI-Based Fall Management Services. The Role of the i-Walker in I-DONTFALL," by Ulises Cortés, Antonio Martínez-Velasco, Cristian Barrué, and Roberta Annicchiarico
- "Syntactic Dependency-Based N-grams as Classification Features," by Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández

The authors of the following papers received the Best Paper Award on the basis of the paper's overall quality, significance, and originality of the reported results:

1st place: "Toward the Creation of Semantic Models Based on Computer-Aided Designs," by Nestor Velasco Bermeo, Miguel González Mendoza, Alexander García Castro, and Irais Heras Dueñas (Mexico, USA)

2nd place: "A New Branch and Bound Algorithm for the Cyclic Bandwidth Problem," by Hillel Romero-Monsivais, Eduardo Rodriguez-Tello, and Gabriel Ramírez (Mexico)

3rd place: "Modelling, Aggregation, and Simulation of a Dynamic Biological System Through Fuzzy Cognitive Maps," by Gonzalo Nápoles, Isel Grau, Maikel León, and Ricardo Grau (Cuba)

The authors of the following paper selected among all papers of which the first author was a full-time student, excluding the papers listed above, received the Best Student Paper Award:

1st place: "Fuzzy Clustering for Semi-Supervised Learning—Case Study: Construction of an Emotion Lexicon," by Soujanya Poria, Alexander Gelbukh, Dipankar Das, and Sivaji Bandyopadhyay (India, Mexico).

In addition, the attendees of the Special Session voted for the following works to receive the Best Poster Award (these papers are included in a separate Special Session proceedings volume):

1st place: “EMG Pattern Recognition System Based on Neural Networks,” by Juan Carlos Gonzalez-Ibarra, Carlos Soubervielle-Montalvo, Omar Vital-Ochoa, and Hector Gerardo Perez-Gonzalez (Mexico)

2nd place: “Conflict Resolution in Multiagent Systems: Balancing Optimality and Learning Speed,” by Aaron Rocha-Rocha, Enrique Munoz de Cote, Saul Pomares Hernandez, and Enrique Sucar Succar (Mexico)

3rd place: “Conception and Implementation of a Supermarket Shopping Assistant System,” by Antonio Marin-Hernandez, Guillermo de Jesús Hoyos-Rivera, Marlon García-Arroyo, and Luis Felipe Marin-Urias (Mexico)

We want to thank all the people involved in the organization of this conference. In the first place, these are the authors of the papers published in this book: it is their research work that gives value to the book and to the work of the organizers. We thank the Track Chairs for their hard work, the Program Committee members and additional reviewers for their great effort spent on reviewing the submissions.

We would like to express our sincere gratitude to the Coordinación para la Innovación y Aplicación de la Ciencia y la Tecnología (CIACyT) of the Universidad Autónoma de San Luis Potosí, CONSYS TELECOM, ACOMEE, Magnetica FM, Faragauss System, Fundación Nikola Tesla, Panadería Penny, and Universidad Politécnica de San Luis Potosí (UPSLP) for their generous support, and in particular to UPSLP for providing the infrastructure for the keynote talks, tutorials, workshops, and technical presentations. We also thank the Oficina de Congresos y Convenciones de San Luis Potosí (OCCSLP) for its valuable effort in organizing the cultural program as well as other logistics activities.

We are deeply grateful to the conference staff and to all members of the Local Committee headed by Omar Montaño Rivas. In particular, we thank Rafael Llamas Contreras, the Logistics Chair, and Gabriela Zárate Rasillo, the OCCSLP representative, for their great effort in resolving logistics issues. Very special thanks go to Liliana Gámez Zavala and Ana María González Ávila for their help in advertizing MICAI and handling all publicity-related issues.

We are in indebted to José Antonio Loyola Alarcón, the Rector of the Universidad Politécnica de San Luis Potosí, for giving us the wonderful opportunity of organizing the conference at this university and for his unconditional support in the organization of the conference in all its stages. We would also like to express our sincere gratitude to the Vice Rector’s Office of the UPSLP headed by Francisco Javier Delgado Rojas; to Igor León O’Farrill, Treasurer General of the UPSLP, and Francisco Cruz Oradaz Salazar, Head of the Facultad de Tecnologías de Información y Telemática, for their warm hospitality.

We gratefully acknowledge support received from the following projects: WIQ-EI (Web Information Quality Evaluation Initiative, European project 269180), PICCO10-120 (ICYT, Mexico City Government), and CONACYT 122030-DST India project “Answer Validation Through Textual Entailment.” The entire submission, reviewing, and selection process, as well as preparation of the proceedings, was supported for free by the EasyChair system (www.easychair.org). Last but not least, we are grateful to Springer for their patience and help in the preparation of this volume.

November 2012

Ildar Batyrshin
Miguel González Mendoza

Organization

Conference Organization

MICAI 2012 was organized by the Mexican Society of Artificial Intelligence (SMIA, Sociedad Mexicana de Inteligencia Artificial) in collaboration with the Universidad Politécnica de San Luis Potosí (UPSLP), Centro de Investigación en Computación del Instituto Politécnico Nacional (CIC-IPN), Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Universidad Nacional Autónoma de México (UNAM), Universidad Autónoma de México (UAM), Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), Universidad Autónoma de Estado de Hidalgo (UAEH), and Instituto Mexicano de Petroleo (IMP).

The MICAI series website is: www.MICAI.org. The website of the Mexican Society of Artificial Intelligence, SMIA, is: www.SMIA.mx. Contact options and additional information can be found on these websites.

Conference Committee

General Chairs	Raúl Monroy Alexander Gelbukh
Program Chairs	Miguel González Mendoza Ildar Batyrshin
Workshop Chair	Alexander Gelbukh
Tutorials Chairs	Felix Castro Espinoza
Keynote Talks Chair	Jesus A. Gonzalez
Financial Chair	Grigori Sidorov
Grant Chairs	Grigori Sidorov Miguel González Mendoza Ildar Batyrshin
Best Thesis Awards Chair	Miguel González Mendoza
Doctoral Consortium Chairs	Oscar Herrera Alcántara Miguel González Mendoza
Organizing Committee Chair	Omar Montaño Rivas
Supervising Ex-President of the SMIA	Carlos Alberto Reyes García

Track Chairs

Natural Language Processing	Sofia Natalia Galicia Haro
Machine Learning and Pattern Recognition	Johan van Horebeek
Data Mining	Felix Castro Espinoza

Intelligent Tutoring Systems	Alexander Gelbukh
Evolutionary and Nature-Inspired Metaheuristic Algorithms	Nareli Cruz Cortés Oliver Schütze
Computer Vision and Image Processing	Oscar Herrera Alcántara
Robotics, Planning and Scheduling	Fernando M. MontesGonzalez
Neural Networks and Hybrid Intelligent Systems	Sergio Ledesma Orozco
Logic, Knowledge-Based Systems	Mauricio Osorio
Multi-Agent Systems, and Distributed AI	Claudia Zepeda Cortés
Fuzzy Systems and Probabilistic Models in Decision Making	Alexander Tulupyev
Bioinformatics and Medical Applications	Jesús A. González

Program Committee

Charan Kumari Achanta
Carlos Acosta
Hector-Gabriel Acosta-Mesa
Ruth Aguilar
Carlos Aguilar Ibáñez
Teresa Alarcón
Alfonso Alba
Rafik Aliev
Leopoldo Altamirano
Gustavo Arroyo
Serge Autexier
J. Gabriel Avina-Cervantes
Victor Ayala-Ramirez
Bogdan Babych
Sivaji Bandyopadhyay
Maria Lucia Barrón-Estrada
Ildar Batyrshin
Hector Becerra
Mokhtar Beldjehem
Leopoldo Bertossi
Albert Bifet
Leticia Cagnina
Felix Calderon
Hiram Calvo
Nicoletta Calzolari
Oscar Camacho Nieto
Jose Luis Carballido
Mario Castelán
Oscar Castillo
Felix Castro Espinoza
Martine Ceberio
Gustavo Cerda-Villafana
Mario Chacon
Niladri Chatterjee
Edgar Chavez
Zhe Chen
Stefania Costantini
Broderick Crawford
Nareli Cruz Cortés
Rolando Cruz Ramírez
Heriberto Cuayahuitl
Iria da Cunha
Oscar Dalmau
Enrique de La Rosa Miranda
Carlos Delgado-Mata
Bernabe Dorronsoro

Beatrice Duval
Asif Ekbal
Michael Emmerich
Alexandr Eremeev
Hugo-Jair Escalante
Boris Escalante Ramírez
Ponciano Jorge Escamilla-Ambrosio
Susana C. Esquivel
Vladimir Estivill-Castro
Julio Cesar Estrada Rico
Gibran Etcheverry
Eugene C. Ezin
Claudia Feregrino
Denis Filatov
Juan J. Flores
Andrea Formisano
Anilu Franco-Arcega
Alfredo Gabaldon
Sofia N. Galicia-Haro
Ana-Gabriela Gallardo-Hernández
Carlos Hugo Garcia Capulin
Ma. de Guadalupe Garcia-Hernandez
Alexander Gelbukh
Arturo Gonzalez
Jesus A. Gonzalez
Miguel Gonzalez Mendoza
José-Joel Gonzalez-Barbosa
Felix F. Gonzalez-Navarro
D. Gutiérrez
Joaquín Gutiérrez Jagüey
Rafael Guzman
Hartmut Haehnel
Jin-Kao Hao
Yasunari Harada
Rogelio Hasimoto
Jean-Bernard Hayet
Antonio Hernandez
Donato Hernández
Arturo Hernandez-Aguirre
Oscar Herrera
Dieter Hutter
Pablo H. Ibarguengoytia
Mario Alberto Ibarra-Manzano
Hugo Jair Escalante
Berend Jan van der Zwaag
Timoleon Kipouros
Ryszard Klempous
Olga Kolesnikova
Vladik Kreinovich
Reinhard Langmann
Adriana Lara
Bruno Lara
Yulia Ledeneva
Sergio Ledesma-Orozco
Yoel Ledo Mezquita
Pierrick Legrand
Guillermo Leguizamón
Eugene Levner
Aurelio Lopez
Omar Lopez
Juan Manuel Lopez Hernandez
Francisco Luna
Gabriel Luque
Tanja Magoc
Luis Ernesto Mancilla
Nashat Mansour
Alexis Marcano Cedeño
J. Raymundo Marcial-Romero
Antonio Marin Hernandez
Luis Felipe Marin Urias
Georgiana Marsic
Luis Martí
Rene Alfredo Martinez Celorio
Edgar Martinez-Garcia
José Fco. Martínez-Trinidad
Jerzy Martyna
Carolina Medina
Patricia Melin
Martin Mendez
Carlos Merida-Campos
Ivan V. Meza
Efrén Mezura-Montes
Mikhail Mikhailov
Gabriela Minetti
Dieter Mitsche
Luís Moniz Pereira
Raul Monroy
Fernando Martin Montes-Gonzalez
Manuel Montes-y-Gómez
Guillermo Morales-Luna

Victor Muñiz
Masaki Murata
Juan Antonio Navarro Perez
Jesús Emeterio Navarro-Barrientos
Juan Carlos Nieves
Sergey Nikolenko
Juan Arturo Nolazco Flores
Leszek Nowak
Iván Olier
Ivan Olmos
Fernando Orduña Cabrera
Felipe Orihuela-Espina
Eber Orozco
Magdalena Ortiz
Mauricio Osorio
Elvia Palacios
Ted Pedersen
Héctor Manuel Pérez
Thierry Peynot
David Pinto
Eunice E. Ponce-de-Leon
Natalia Ponomareva
Volodymyr Ponomaryov
Edgar Alfredo Portilla-Flores
Joel Quintanilla
Julio Cesar Ramos
Carolina Reta
Alberto Reyes
Orion Fausto Reyes-Galaviz
Carlos A. Reyes-Garcia
Jean-Michel Richer
Mariano Rivera
Antonio Robles-Kelly
Erik Rodner
Arles Rodriguez
Eduardo Rodriguez-Tello
Katya Rodriguez-Vazquez
Leandro Fermín Rojas Peña
Paolo Rosso
Horacio Rostro Gonzalez
Jianhua Ruan
Imre Rudas
Salvador Ruiz Correa
Jose Ruiz-Pinales
Leszek Rutkowski
Andriy Sadovnychyy
Carolina Salto
Gustavo Sanchez
Guillermo Sanchez-Diaz
Abraham Sánchez López
Raul E. Sanchez-Yanez
Jose Santos
Oliver Schütze
Nikolay Semenov
Pinar Senkul
Shahnaz N. Shahbazova
Grigori Sidorov
Gerardo Sierra
Alexander Sirotkin
Peter Sosnin
Humberto Sossa Azuela
Marta Rosa Soto
Ramon Soto
Ricardo Soto
Juan-Manuel Torres-Moreno
Heike Trautmann
Leonardo Trujillo
Alexander Tulupyev
Johan van Horebeek
Massimiliano Vasile
Francois Vialatte
Javier Vigueras
Manuel Vilares Ferro
Andrea Villagra
Toby Walsh
Cornelio Yáñez-Márquez
Ramon Zatarain
Claudia Zepeda Cortes

Additional Reviewers

Mehran Ahmadlou
Israel Becerra
Eduardo Cabal
Nina Dethlefs
Elva Diaz
Christian Dominguez Medina
Milagros Fernández Gavilanes
Benjamin Hernandez
Md. Jamiul Jahid
Rafael Murrieta
José Carlos Ortiz-Bayliss
Partha Pakray
Obdulia Pichardo
Soujanya Poria
Francisco Jose Ribadas-Pena
Edgar Rodriguez
Jose Rodriguez
Octavio Augusto Sánchez Velázquez
José Arturo Tejeda-Gómez

Organizing Committee

Local Chair	Omar Montaño Rivas
Logistics Chairs	Rafael Llamas Contreras
	Cesar Guerra
Registration Chairs	Hugo Gonzalez
	Victor Fernández
Sponsorship Chair	Jorge Simón
Publicity Chair	Liliana Gámez
Financial Chair	Mario Alberto Martínez

Table of Contents – Part II

Natural Language Processing

Invited paper

Evolutionary and Nature-Inspired Metaheuristic Algorithms

Neural Networks and Hybrid Intelligent Systems

Fuzzy Systems and Probabilistic Models in Decision Making

Table of Contents – Part I

Machine Learning and Pattern Recognition

Invited Paper

Computer Vision and Image Processing

Robotics

Knowledge Representation, Reasoning, and Scheduling

Medical Applications of Artificial Intelligence

Invited Paper

Syntactic Dependency-Based N-grams as Classification Features

Grigori Sidorov[1], Francisco Velasquez[1], Efstathios Stamatatos[2],
Alexander Gelbukh[1], and Liliana Chanona-Hernández[3]

[1] Center for Computing Research (CIC),
Instituto Politécnico Nacional (IPN), Mexico City,
Mexico
[2] University of the Aegean,
Greece
[3] ESIME, Instituto Politécnico Nacional (IPN), Mexico City,
Mexico
www.cic.ipn.mx/~sidorov

Abstract. In this paper we introduce a concept of syntactic n-grams (sn-grams). Sn-grams differ from traditional n-grams in the manner of what elements are considered neighbors. In case of sn-grams, the neighbors are taken by following syntactic relations in syntactic trees, and not by taking the words as they appear in the text. Dependency trees fit directly into this idea, while in case of constituency trees some simple additional steps should be made. Sn-grams can be applied in any NLP task where traditional n-grams are used. We describe how sn-grams were applied to authorship attribution. SVM classifier for several profile sizes was used. We used as baseline traditional n-grams of words, POS tags and characters. Obtained results are better when applying sn-grams.

Keywords: syntactic n-grams, sn-grams, parsing, classification features, syntactic paths, authorship attribution.

1 Introduction

N-gram based techniques are predominant in modern NLP and its applications. Traditional n-grams are sequences of elements as they appear in texts. These elements can be words, characters, POS tags, or any other elements as they encounter one after another. Common convention is that "n" corresponds to the number of elements in the sequence.

The main idea of this paper is that n-grams can be obtained based on the order in which the elements are presented in syntactic trees. Namely, we follow a path in the tree and construct n-grams, rather than taking them as they appear in the surface representation of the text. Thus, we consider as neighbors the words (or other elements like POS tags, etc.) that follow one another in the path of the syntactic tree, and not in the text. We call such n-grams **syntactic n-grams** **(sn-grams)**. The great advantage of sn-grams is that they are based on syntactic relations of words and, thus,

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 1–11, 2013.

each word is bound to its “real” neighbors, ignoring the arbitrariness that is introduced by the surface structure.

For example, let us consider two phrases: “*eat with wooden spoon*” vs. “*eat with metallic spoon*”, see Fig. 1. Note that we can use both dependency and constituency representations of syntactic relations. They are equivalent if the head of each constituent is known. In our example of constituents, the heads are marked by heavier lines. It is very easy to add information about heads into constituency based grammar, namely, one of the components should be marked as the head in the rules.

In case of dependencies, we follow the path marked by the arrows and obtain sn-grams. In case of constituents we first “promote” the head nodes so that they occupy the places in bifurcations, as shown in Fig. 2. Then we obtain sn-grams starting from the dependent constituencies and taking the heads from the bifurcations.

Let us consider the case of bigrams for this example. If we extract traditional bigrams from the phrases, then they have only one bigram in common: “*eat with*”. Meanwhile, if we consider sn-grams, then two common bigrams are found: “*eat with*”, “*with spoon*”.

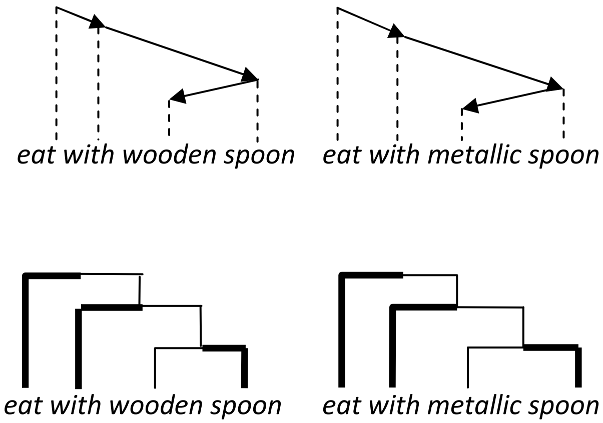


Fig. 1. Representations of syntactic relations

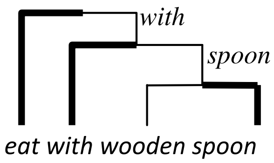


Fig. 2. Promoted head nodes

The same happens in case of trigrams. In case of traditional n-grams, there are no common n-grams, but if we use sn-grams then there is one common trigram: "*eat with spoon*".

In this case, sn-grams allow ignoring the surface phenomenon of the English language that adds an adjective before the noun and in this way spoils traditional bigrams/trigrams. The same happens in case of, say, subordinate clauses, and, in general, in any type of syntactic structures.

Other quite obvious possibility is to construct sn-grams ignoring auxiliary words (stop words). We shall follow the paths in the tree and just pass through the nodes of the stop words. In case of our examples, they have the common sn-bigram "*eat spoon*" that will not be obtained using traditional n-grams.

Sn-grams can be used as classification features in the same manner as traditional n-grams. So, in any task in which we use traditional n-grams, we can apply sn-grams as well.

Obvious problem with sn-grams is that syntactic parsing is required. The parsing can take considerable time and there is a problem of availability of parsers for particular languages, though for well-studied languages like English or Spanish this consideration is not a problem.

In this paper, we apply sn-grams in authorship attribution problem. We conducted experiments that obtain better results for this task with sn-grams than with traditional n-grams. Besides, we believe that sn-grams have real linguistic interpretation as far as the writing style of authors is concerned because they reflect real syntactic relations.

Further in this paper, we first, discuss syntactic n-grams (Section 2), present relevant work for authorship attribution (Section 3), and then present experimental results comparing the proposed approach with traditional methods (Section 4). Finally, we summarize the conclusions drawn by this study.

2 Syntactic n-grams (sn-grams)

The advantage of syntactic n-grams (sn-grams), i.e., n-grams that are constructed using paths in syntactic trees, is that they are less arbitrary than traditional n-grams. Thus, their number is less than the number of traditional n-grams. Besides, they can be interpreted as linguistic phenomenon, while traditional n-grams have no plausible linguistic interpretation—they are merely statistical artifact.

The justification of the idea of sn-grams is related to introduction of linguistic information into statistically based methods. We believe that this idea helps to overcome the main disadvantage of the traditional n-grams—they contain many arbitrary elements, i.e., a lot of noise.

The obvious disadvantage of syntactic n-grams is that previous syntactic processing is necessary. This consumes significant time and it is not easy to apply to some languages, because a syntactic parser and a set of lexical resources that are used by the parser are needed and not for any language these resources are available.

Previously, similar ideas were related to some specific tasks like using additional syntactic information in machine translation [1] or generation in machine translation

[2], without the generalization and taxonomy that we propose in this paper. The term syntactic n-gram is not very common and its importance is underestimated. It is used, for example, in [3] for extraction of polarity of syntactic constituents (chunks) as a whole element.

Note that there are attempts to overcome the disadvantages of traditional n-grams using purely statistical approaches. We should mention skip-grams and Maximal Frequent Sequences (MFS).

Skip-grams are very similar to n-grams, but during their construction some elements of the corresponding sequence are ignored (skipped). It is an attempt as well to avoid possible noise, namely, by considering random variations in texts. There can be skips with various skip steps.

An example of skip-grams: say, for the sequence ABCDE, we obtain traditional bigrams AB, BC, CD, and DE. Skip-bigrams with skip step of 1 will be AC, DB, and CE. Various skip steps can be used. Usually skip-grams also include traditional n-grams, in which case the skip step is zero. The problem with skip-grams is that their number grows very fast.

Maximal Frequent Sequences (MFS) [4] are skip-grams with major frequency, i.e., we take into account only skip-grams whose frequencies are greater than a certain threshold. The problem with MFS is that for their construction complex algorithms should be used and it takes substantial processing time. Another disadvantage of MSF is that unlike sn-grams, they depend on text collection. And similar to skip-grams, no linguistic interpretation of MFS is possible in general case.

According to the **types of elements** that form syntactic n-grams, there can be various types of sn-grams:

- Word sn-grams: the elements of sn-gram are words,
- POS sn-grams: the elements of sn-gram are POS tags,
- Sn-grams of syntactic relations tags: SR tags, the elements of sn-gram are names of syntactic relations,
- Mixed sn-grams: sn-grams are composed by mixed elements like words (lexical units), POS tags and/or SR tags. Since a sn-gram follows a syntactic link, then there are reasons to use mixed sn-grams, for example, they can reflect subcategorization frames. There are many possible combinations regarding to what part—the head word or the dependent word in the relation—should be represented by lexical unit, POS tag, or SR tag. These combinations should be explored experimentally in future.

Note that sn-grams of characters are impossible.

As far as the **treatment of stop words** is concerned, they can be ignored or taken into account, as we mentioned before.

3 Sn-grams of SR Tags

For the experiments described in this paper, we used syntactic relations (SR tags) as elements of sn-grams. Stanford parser was used for determining SR tags, POS tags, and for construction of dependency-based syntactic trees. Although the parsing process was time consuming for a large corpus, it was performed only once, so the subsequent experiments do not take substantial time.

Let us consider an example sentence "*Economic news have little effect on financial markets*" that is used in some NLP courses. Stanford parser output for dependency relations is:

nn (news-2, Economic-1)
nsubj (have-3, news-2)
root (ROOT-0, have-3)
amod (effect-5, little-4)
dobj (have-3, effect-5)
prep (effect-5, on-6)
amod (markets-8, financial-7)
pobj (on-6, markets-8)

This representation contains the following information: relation name (shown at the beginning of each line) and related words in parenthesis, where the first argument within the parentheses represents the head and the second represents the dependent element. Thus, *amod (effect-5, little-4)* means that there is a relation of adjectival modifier (*amod*) from *effect* to *little*. The numbers correspond to word numbers in the sentence. Stanford parser handles 53 relations.

SR tags are presented in squares over the syntactic tree in Fig. 3.

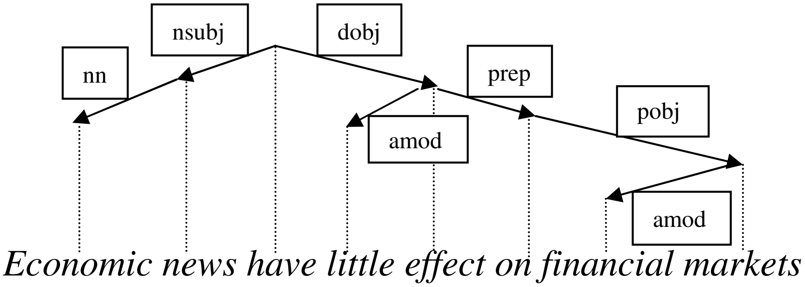


Fig. 3. Dependencies tree of the example

Based on the relations established by the parser, we obtain sn-grams following the arrows. For example, from this tree we can extract the following sn-grams:

nsubj → *nn*
dobj → *amod*
dobj → *prep* → *pobj* → *amod*

Note that we use all possible paths, for example, if we consider trigrams, then the paths *dobj*→*prep*→*pobj* and *prep* → *pobj* → *amod* will be extracted, etc.

4 Relevant Work on Authorship Attribution

Authorship attribution deals with an old and difficult question: how to assign a text of unknown or disputed authorship to a member of a set of candidate authors for whom undisputed samples of texts are available [8]. Despite its application to literary works, the rapid expansion of online text in Internet media (e.g., blogs, e-mail messages, social media postings, etc.) revealed practical applications of authorship attribution usually associated with forensic tasks [11].

The automated approaches to this problem involve the use of statistical or machine learning methods [6]. From the machine learning point of view, authorship attribution can be seen as a single-label multi-class classification task [10]. There are two basic steps: first, the texts should be appropriately represented as vectors of numerical values and, then, a classification algorithm can use these vectors to estimate the probability of class membership for any given text.

Thousands of stylometric features have been proposed so far. These features can be distinguished in the following main categories according to the textual analysis they require [6]: lexical features (e.g., function word frequencies, word n-gram frequencies, vocabulary richness measures, etc.), character features (e.g., letter frequencies and character n-grams), syntactic features (e.g., part-of-speech tag frequencies, sentence and phrase structure measures, rewrite rule frequencies etc.), semantic features (e.g., synonym measures, semantic dependency measures etc.), and application-specific features (e.g., font size, font color, specific word frequencies, etc.). Until now, a number of studies [13, 14, 18] have shown that the most effective measures are lexical and character features.

Note that information about syntactic relations usually is not used in this task, being one of the exceptions [1], where the authors analyze syntactic rewriting rules. In this paper we show that syntactic relations taken as sn-grams represent very effective measure.

In this paper, we use as baseline methods character features, lexical features (words), and POS tags obtained using traditional n-grams, i.e., according to the appearance of the elements in texts.

5 Experimental Results and Discussion

The experiments were performed over the corpus data. The corpus used in our study includes texts downloaded from the Project Gutenberg. We selected books of native English speaking authors that have literary production in a similar period. In this paper, all experiments are conducted for the corpus of 39 documents by three authors.

For evaluation of the experiments, we used 60% of the data for training, and the rest 40% for classification, as presented in Table 1.

We used WEKA implementation of Support Vector Machines (SVM) classifier. SVM is known to produce very good results in the task of authorship attribution.

We used several features as baseline: word based features, character based features, and POS tags. For baseline features, we used traditional n-gram technique, i.e., the elements are taken as they appear in the texts. We applied sn-grams technique for the same corpus with the results that outperform baseline methods.

Table 1. Training and classification data

	Training		**Classification**	
Author	Novels	Size (MB)	Novels	Size (MB)
Booth Tarkington	8	3.6	5	1.8
George Vaizey	8	3.8	5	2.1
Louis Tracy	8	3.6	5	2.2
Total	**24**	**11**	**15**	**6.1**

We use the term "profile size" for representing the first most frequent n-grams/sn-grams, e.g., for profile size of 400 only first 400 most frequent n-grams are used. We tested various thresholds for profile size and selected five thresholds for profile size as presented in all tables with the results.

When the table cell contains *NA (not available)*, it means that our data were insufficient to obtain the corresponding number of n-grams. It happens only with bigrams because in general there are less bigrams than trigrams, etc. In these cases the total number of all bigrams is less than the given profile size.

In Tables 2, 3 and 4 the results of the baseline methods are presented for the selected baseline methods. Table 5 contains the results obtained using sn-grams. For better appreciation of the comparison of the results, we present Tables 6 to 9, where the results are grouped by the size of n-grams/sn-grams.

Table 2. Word based n-grams (baseline)

Profile size	**n-gram size**			
	2	**3**	**4**	**5**
400	86%	81%	67%	45%
1,000	86%	71%	71%	48%
4,000	86%	<u>95%</u>	67%	48%
7,000	86%	90%	71%	45%
11,000	89%	90%	75%	33%

Table 3. Character based n-grams (baseline)

Profile size	**n-gram size**			
	2	**3**	**4**	**5**
400	90%	76%	81%	81%
1,000	95%	86%	86%	76%
4,000	90%	95%	90%	86%
7,000	NA	90%	86%	86%
11,000	NA	<u>100%</u>	90%	86%

Table 4. n-grams based on POS tags (baseline)

Profile size	n-gram size			
	2	**3**	**4**	**5**
400	90%	90%	76%	62%
1,000	95%	90%	86%	67%
4,000	NA	100%	86%	86%
7,000	NA	100%	90%	86%
11,000	NA	95%	90%	86%

Table 5. sn-grams based on SR tags

Profile size	n-gram size			
	2	**3**	**4**	**5**
400	100%	100%	87%	93%
1,000	100%	100%	87%	93%
4,000	100%	100%	93%	73%
7,000	100%	100%	87%	87%
11,000	100%	100%	93%	87%

In the following tables we present the data grouped by n-gram sizes.

Table 6. Comparison for bigrams

Profile size	Features			
	sn-grams of SR tags	**n-grams of POS tags**	**Character based n-grams**	**Word based n-grams**
400	100%	90%	90%	86%
1,000	100%	95%	95%	86%
4,000	100%	NA	90%	86%
7,000	100%	NA	NA	86%
11,000	100%	NA	NA	89%

Table 7. Comparison for trigrams

Profile size	Features			
	sn-grams of SR tags	n-grams of POS tags	Character based n-grams	Word based n-grams
400	100%	90%	76%	81%
1,000	100%	90%	86%	71%
4,000	100%	100%	95%	95%
7,000	100%	100%	90%	90%
11,000	100%	95%	100%	90%

Table 8. Comparison for 4-grams

Profile size	Features			
	sn-grams of SR tags	n-grams of POS tags	Character based n-grams	Word based n-grams
400	87%	76%	81%	67%
1,000	87%	86%	86%	71%
4,000	93%	86%	90%	67%
7,000	87%	90%	86%	71%
11,000	93%	90%	90%	75%

Table 9. Comparison for 5-grams

Profile size	Features			
	sn-grams of SR tags	n-grams of POS tags	Character based n-grams	Word based n-grams
400	93%	62%	81%	45%
1,000	93%	67%	76%	48%
4,000	73%	86%	86%	48%
7,000	87%	86%	86%	45%
11,000	87%	86%	86%	33%

It can be appreciated that in all cases sn-gram technique outperforms the technique based on traditional n-grams. We consider that SR tags and POS tags are similar for the purposes of our comparison; both are small sets of tags: 36 and 53 elements, associated with words. Note that the majority of these elements have very low frequency.

As can be seen, topline of our task is very high: 100%. It is related to the fact that we use much data and our classification only distinguishes between three classes. In some cases, baseline methods also reach the topline. Still, it happens only for a small number of specific profile sizes. The best results are obtained by sn-grams using bigrams and trigrams for any profile size. For any combination of parameters baseline methods got worse results than sn-grams.

The question can arise if it is worth working with the small number of classes. In our opinion, it is useful and important. First of all, authorship attribution is often a real world application in case of a dispute over the authorship of a document, and in this case the number of classes is reduced to two or three, i.e., it is our situation.

6 Conclusions

In this paper we introduce a concept of syntactic n-grams (sn-grams). The difference between traditional n-grams and sn-grams is related to the manner of what elements are considered neighbors. In case of sn-grams, the neighbors are taken by following syntactic relations in syntactic trees, while traditional n-grams are formed as they appear in texts.

Any syntactic representation can be used for application of sn-gram technqiue: dependency trees or constituency trees. In case of dependency tree, we should follow the syntactic links and obtain sn-grams. In case of constituency trees, some additional steps should be made, but these steps are very simple.

Sn-grams can be applied in any NLP task where traditional n-grams are used.

We conducted experiments for authorship attribution task using SVM for several profile sizes. Relatively large corpus of works of three authors was used.

We used as baseline traditional n-grams of words, POS tags and characters. The results show that sn-gram technique outperforms the baseline technique.

As future work, we will apply the idea of sn-grams to other NLP tasks. As far as our particular data is concerned, we will perform the exhaustive comparison of all features using technique of sn-grams. We also plan to experiment with various types of mixed sn-grams.

Acknowledgements. Work done under partial support of Mexican government (projects CONACYT 50206-H and 83270, SNI) and Instituto Politécnico Nacional, Mexico (projects SIP 20111146, 20113295, 20120418, COFAA, PIFI), Mexico City government (ICYT-DF project PICCO10-120) and FP7-PEOPLE-2010-IRSES: Web Information Quality - Evaluation Initiative (WIQ-EI) European Commission project 269180. We also thank Sabino Miranda and Francisco Viveros for their valuable and motivational comments.

References

1. Khalilov, M., Fonollosa, J.A.R.: N-gram-based Statistical Machine Translation versus Syntax Augmented Machine Translation: comparison and system combination. In: Proceedings of the 12th Conference of the European Chapter of the ACL, pp. 424–432 (2009)
2. Habash, N.: The Use of a Structural N-gram Language Model in Generation-Heavy Hybrid Machine Translation. In: Belz, A., Evans, R., Piwek, P. (eds.) INLG 2004. LNCS (LNAI), vol. 3123, pp. 61–69. Springer, Heidelberg (2004)
3. Agarwal, A., Biads, F., Mckeown, K.R.: Contextual Phrase-Level Polarity Analysis using Lexical Affect Scoring and Syntactic N-grams. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL), pp. 24–32 (2009)
4. García-Hernández, R.A., Martínez Trinidad, J.F., Carrasco-Ochoa, J.A.: Finding Maximal Sequential Patterns in Text Document Collections and Single Documents. Informatica 34(1), 93–101 (2010)
5. Baayen, H., Tweedie, F., Halteren, H.: Outside The Cave of Shadows: Using Syntactic Annotation to Enhance Authorship Attribution. Literary and Linguistic Computing, 121–131 (1996)
6. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)
7. Holmes, D.I.: The evolution of stylometry in humanities scholarship. Literary and Linguistic Computing 13(3), 111–117 (1998)
8. Juola, P.: Authorship Attribution. Foundations and Trends in Information Retrieval 1(3), 233–334 (2006)
9. Juola, P.: Ad-hoc authorship attribution competition. In: Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, pp. 175–176 (2004)
10. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1) (2002)
11. Abbasi, A., Chen, H.: Applying authorship analysis to extremist-group web forum messages. IEEE Intelligent Systems 20(5), 67–75 (2005)
12. van Halteren, H.: Author verification by linguistic profiling: An exploration of the parameter space. ACM Transactions on Speech and Language Processing 4(1), 1–17 (2007)
13. Grieve, J.: Quantitative authorship attribution: An evaluation of techniques. Literary and Linguistic Computing 22(3), 251–270 (2007)
14. Luyckx, K.: Scalability Issues in Authorship Attribution. Ph.D. Thesis, University of Antwerp (2010)
15. Argamon, S., Juola, P.: Overview of the international authorship identification competition at PAN-2011. In: 5th Int. Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (2011)
16. Diederich, J., Kindermann, J., et al.: Authorship attribution with support vector machines. Applied Intelligence 19(1), 109–123 (2003)
17. Escalante, H., Solorio, T., et al.: Local histograms of character n-grams for authorship attribution. In: 49th Annual Meeting of the Association for Computational Linguistics, pp. 288–298 (2011)
18. Keselj, V., Peng, F., et al.: N-gram-based author profiles for authorship attribution. Computational Linguistics 3, 225–264 (2003)
19. Koppel, M., Schler, J., et al.: Authorship attribution in the wild. Language Resources and Evaluation 45(1), 83–94 (2011)
20. Koppel, M., Schler, J., et al.: Measuring differentiability: unmasking pseudonymous authors. Journal of Machine Learning Research, 1261–1276 (2007)

Semantic Annotation for Textual Entailment Recognition

Assaf Toledo[1], Sophia Katrenko[1], Stavroula Alexandropoulou[1], Heidi Klockmann[1], Asher Stern[2], Ido Dagan[2], and Yoad Winter[1]

[1] Utrecht University, Utrecht, The Netherlands
[2] Bar Ilan University, Ramat Gan, Israel

Abstract. We introduce a new semantic annotation scheme for the Recognizing Textual Entailment (RTE) dataset as well as a manually annotated dataset that uses this scheme. The scheme addresses three types of modification that license entailment patterns: *restrictive*, *appositive* and *conjunctive*, with a formal semantic specification of these patterns' contribution for establishing entailment. These inferential constructions were found to occur in 77.68% of the entailments in the RTE 1-3 corpora. They were annotated with cross-annotator agreement of 70.73% on average. A central aim of our annotations is to examine components that address these phenomena in RTE systems. Specifically, the new annotated dataset is used for examining a syntactic rule base within the BIUTEE recognizer, a publicly available entailment system. According to our tests, the rule base is rarely used to process the phenomena annotated in our corpus and most of the recognition work is done by other components in the system.[1]

Keywords: RTE Textual Entailment Semantic Annotation.

1 Introduction

The *Recognizing Textual Entailment* (RTE) [9] task aims to automatically determine whether an entailment relation obtains between a naturally occurring **text** (T) and a **hypothesis** sentence (H). In the RTE corpus, which is currently the only available textual entailments resource, the entailment candidates are marked manually as valid/invalid.[2] This categorization contains no indication of the linguistic and informational processes that underlie entailment. Consider the following example from RTE 2:

– T: The widow of John Lennon, Yoko Ono, may take legal action against a new breakfast cereal called "Strawberry Fields" which she believes is too close in name to Lennon's song.
– H: Yoko Ono is John Lennon's widow.

[1] The annotated corpus is freely downloadable.

[2] Pairs of sentences in RTE 1-3 are categorized in two classes: *yes-* or *no-entailment*; the data in RTE 4-5 are categorized in three classes: *entailment*, *contradiction* and *unknown*. We label the judgments *yes-entailment* from RTE 1-3 and *entailment* from RTE 4-5 as *valid*, and the other judgments as *invalid*.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 12–25, 2013.

Here, the judgment on the validity of this entailment is based on the semantic properties of the appositive construction - *The widow of John Lennon, Yoko Ono* - whereby two noun phrases refer to the same entity.

Because linguistic information of this kind is not analyzed or even stated in the RTE corpus, it is difficult to develop new entailment recognizers or evaluate existing ones based on theoretical research that has been carried out in natural-language semantics or on supervised learning techniques. For previous work that yielded similar conclusions, see [28].

In this paper we describe a new scheme for annotating valid entailments in the RTE corpus and provide an RTE dataset annotated manually using this scheme.[3] The scheme addresses three types of modification that license entailment: *restrictive*, *appositive* and *conjunctive*; a formal semantic specification is provided for each of these, as well as for its role in establishing the entailment [24,27,29,10]. The annotation proposed in this paper is *semantic* in the sense that it targets semantic relations that govern the inferences licensed in various expressions, rather than merely annotating the syntax of these expressions.

The rationale behind our decision to focus on the three semantic phenomena listed above is that they are common in the RTE dataset and linguistically intuitive enough to yield high inter-annotator agreement. The annotated corpus is designated to serve as a benchmark for a theoretically informed evaluation of entailment modules in RTE systems. In addition, it can be used in the development of entailment recognizers by facilitating automatic learning of the above constructions.

The results of this work show that, in the corpora of RTE 1-3, the inferential constructions marked according to the proposed scheme occur in 77.68% of the valid entailments, with an average inter-annotator agreement of 70.73% (see Section 3.3). As a use case for our corpus, we examined the processing of the annotated constructions by BIUTEE [30], a state of the art entailment recognizer which employs a rule base targeting a broad range of inferential patterns.[4] We found that, in its treatment of RTE 1-3, BIUTEE processes only 4.52% of the entailment cases based on rules that correspond to the annotated constructions. It seems, therefore, that the rule base examined rarely participates in the processing of the phenomena investigated.

The structure of this paper is as follows. Section 2 elucidates the connection between this and previous annotation work on the RTE data. In Section 3 we introduce the annotation scheme, elaborate on the methods employed, and present some quantitative data on the targeted constructions and inter-annotator agreement. Section 4 describes a use case of the annotated corpus with reference to the rule base employed by the BIUTEE recognizer. Section 5 is a conclusion.

[3] The annotated corpus is available at: `http://logiccommonsense.wp.hum.uu.nl/papers/annotatedrte`

[4] `http://cs.biu.ac.il/~nlp/downloads/biutee`; version 2.4.1.

2 Related Work

During the past few years, several methodological proposals and annotation projects have been attempted with the purpose of uncovering and documenting entailment processes in the RTE. The main assumption underlying most of the work in this direction is that decomposing the complex entailment problem would improve the performance of RTE systems.

In [5], a methodology is described for creating specialized entailment data sets by isolating linguistic phenomena relevant for entailment. Pairs of text and hypothesis from the existing corpus were used to generate a set of mono-thematic pairs, each containing one specific phenomenon that takes part in the original entailment. As part of a feasibility study, this methodology was applied to a sample of 90 pairs randomly extracted from RTE 5. The phenomena were divided into broad categories (e.g. lexical, syntactic, etc.) and then into more fine-grained categories (e.g. synonymy, active-passive, etc.), and a general rule of inference was defined for each of these categories (e.g. *argument realization*: "x's y" → "y of x").

This methodology allows a detailed analysis of the entailments in the corpus, but a full analysis of all entailment patterns in the corpus would necessarily involve complex judgments, and this, in turn, would make high cross-annotator consistency very hard to achieve. Moreover, as discussed in Section 3.3, our experience shows that efficient annotation with high cross-annotator consistency is hard to obtain even in more restricted cases which involve less complex judgments.

Our annotation work is to a large extent in line with the proposal described in [28], whose authors appeal to the NLP community to contribute to entailment recognition work by incrementally annotating the phenomena that underlie the inferences in the RTE corpus. However our annotation scheme does not require a full understanding of the phenomena involved in the entailment in a given T-H pair. Nor does it aim to uncover all the connections between these phenomena, and accordingly, it does not specify the order by which the inferential steps are carried out in each particular case.

3 Corpus Description

This section details our annotation strategy, the inference principles it is based on, the annotation work-flow, and some quantitative data on the targeted constructions and inter-annotator agreement.

3.1 Phenomena Annotated

The choice of phenomena for annotation is based on the following criteria:

1. Phenomena that are commonly involved in entailments.
2. Phenomena that are well understood in the semantic literature and that lend themselves readily to linguistic intuitions as well as to an analysis that is likely to yield high annotation consistency.
3. Phenomena that do not require sophisticated abstract representations and which therefore are easy to classify.

Based on these considerations, three types of modifications were selected for annotation: restrictive modification, appositive modification and intersective modification (conjunction).

An important feature of our annotation is that it marks inferences by aligning strings in the text and the hypothesis. This is done by pairing each annotation in the text with a corresponding annotation in the hypothesis that marks the output of the inferential process of the phenomenon in question. In the next subsections we describe the annotated phenomena in detail and in each example we underline the annotated part in the text with its correspondence in the hypothesis.

Restrictive Modification (RMOD). RMOD is an instance of two adjacent expressions in which one expression (the *modifier*) restricts the semantic class of objects denoted by the other (the *modifiee*). RMOD licenses an entailment pattern of modifier subsumption:

– T: *A Cuban*$_{Modifier}$ *American*$_{Modifiee}$ *who is accused of espionage pleads innocent.*
– H: *American accused of espionage.*

In this case, *Cuban* modifies *American* and restricts the set of Americans to Cuban Americans. This instance of RMOD validates the inference from *Cuban Americans* to *Americans* which is required for establishing the entailment.

The following examples contain additional syntactic configurations in which RMOD licenses inferences:

– A verb phrase restricted by a prepositional phrase:
  - T: *The watchdog International Atomic Energy Agency meets in Vienna*$_{Modifiee}$ *on September 19*$_{Modifier}$.
  - H: *The International Atomic Energy Agency holds a meeting in Vienna.*

– A noun phrase restricted by a prepositional phrase:
  - T: *U.S. officials have been warning for weeks of possible terror attacks*$_{Modifiee}$ *against U.S. interests*$_{Modifier}$.
  - H: *The United States has warned a number of times of possible terrorist attacks*.

Appositive Modification (APP). Apposition involves two adjacent expressions whereby one part designates an entity and the other supplements its description by additional information. APP licenses three main entailment patterns:

- Appositive subsumption (left part):
  - T: *Mr. Conway, Iamgold's chief executive officer, said the vote would be close.*
  - H: *Mr. Conway said the vote would be close.*
- Appositive subsumption (right part):
  - T: *The country's largest private employer, Wal-Mart Stores Inc., is being sued by a number of its female employees who claim they were kept out of jobs in management because they are women.*
  - H: *Wal-Mart sued for sexual discrimination.*
- Identification of the two parts of the apposition as referring to one another:
  - T: *The incident in Mogadishu, the Somali capital, came as U.S. forces began the final phase of their promised March 31 pullout.*
  - H: *The capital of Somalia is Mogadishu.*

In addition to appositive constructions as illustrated above, APP appears in several more syntactic constructions:

- Non-Restrictive Relative Clauses:
  - T: *A senior coalition official in Iraq said the body, which was found by U.S. military police west of Baghdad, appeared to have been thrown from a vehicle.*
  - H: *A body has been found by U. S. military police.*
- Title Constructions:
  - T: *Prime Minister Silvio Berlusconi was elected March 28 with a mandate to reform Italy's business regulations and pull the economy out of recession.*
  - H: *The Prime Minister is Silvio Berlusconi.*

Conjunction (CONJ). CONJ is an instance of two or more adjacent expressions that are interpreted intersectively. Typically CONJ licenses an entailment pattern of conjunct subsumption:

- T: *Nixon was impeached and became the first president ever to resign on August 9th 1974.*
- H: *Nixon was the first president ever to resign.*

In this example the conjunction intersects the two verb phrases *was impeached* and *became the first president ever to resign.* The entailment relies on subsumption of the full construction to the second conjunct.

In addition to canonical conjunctive constructions, CONJ appears also in Restrictive Relative Clauses whereby the relative clause is interpreted intersectively with the noun being modified:

- T: *Iran will soon release eight British servicemen detained along with three vessels.*
- H: *British servicemen detained.*

3.2 Marking Annotations

Given a pair from the RTE in which the entailment relation obtains between the text and hypothesis, the task for the annotators is defined as follows:

1. Read the data and verify the entailment.
2. Describe informally why the entailment holds.
3. Annotate all the instances of the phenomena described in Section 3.1 that play a role in the inferential process.

The annotations were performed using GATE Developer [8] and recorded above the original RTE XML files. The annotators use GATE annotation schemes that were defined to correspond to RMOD, APP and CONJ as shown in Table 1. [5]

Table 1. GATE Annotation Schemes

Phenomenon	Annotation Schemes
RMOD	r_modification
APP	apposition, title, rel_clause
CONJ	conjunction, rel_clause

The work was performed in two steps: (1) marking the relevant string in the text using one of GATE annotation schemes that had been defined for the purpose (e.g. *apposition*), and (2) - marking a string in the hypothesis that corresponds to the output of the inferential process. The annotation in the hypothesis is done using a dedicated *reference_to* scheme.

Example

Consider the following pair from RTE 2:

- T: *The anti-terrorist court found two men guilty of murdering Shapour Bakhtiar and his secretary Sorush Katibeh, who were found with their throats cut in August 1991.*
- H: *Shapour Bakhtiar died in 1991.*

The entailment patterns in this example can be explained by appealing to the semantics of APP, CONJ and RMOD, as follows:

- APP: The appositive modification in *Shapour Bakhtiar and his secretary Sorush Katibeh, who were found with their throats cut in August 1991* licenses the inference that *Shapour Bakhtiar and his secretary Sorush Katibeh were found with their throats cut in August 1991.*

[5] The scheme *rel_clause* appears twice in this table because it is used for annotating non-restrictive relative clauses, expressing appositive modification (APP) and also restrictive relative clauses, expressing intersective modification (CONJ). The phenomena APP and CONJ are annotated using several annotation schemes in order to capture the different syntactic expressions that they allow.

- RMOD: The restrictive modification in *August 1991* licenses a subsumption of this expression to *1991*.
- CONJ: The conjunction in *Shapour Bakhtiar and his secretary Sorush Katibeh* licenses a subsumption of this expression to *Shapour Bakhtiar*.

By combining these three patterns, we can infer that *Shapour Bakhtiar was found with his throat cut in 1991*.[6] In figure 1 we show the annotation tags of CONJ which were added in the text body and serve as boundary markers for the phenomenon and as pointers to the annotation content tags. Figure 2 presents the annotation content tags that detail the conjunction.

- Text:
  The anti-terrorist court found two men guilty of murdering ***<Node id= “11404”/>Shapour Bakhtiar and his secretary Sorush Katibeh <Node id=“11453”/>****, who were found with their throats cut in August 1991.*
- Hypothesis:
  <Node id=“11511”/>Shapour Bakhtiar <Node id=“11527”/> *died in 1991.*

Fig. 1. Annotation tags in the text body

3.3 Annotation Statistics

The annotated corpus is based on the scheme described above, applied to the datasets of RTE 1-4 [9,2,14,13]. The statistics in Table 2 are based on analyzing the annotation work of RTE 1-3 (development and test sets).

We performed two cross-annotator consistency checks. In each check we picked a number of entailment pairs that both annotators worked on independently and compared the phenomena that they annotated. We reached cross-annotator consistency on 70.73% of the annotations on average, as reported in Table 3.

4 Corpus Use Case

Our annotations reflect human judgments on phenomena that take part in inferential processes in the RTE corpus. A straightforward use case of these annotations is to examine the coverage of the annotated phenomena by specific modules that are designed to address them in entailment recognizers. For clarity, we treat these modules of recognizers as rule bases that assign a rule - an inferential operation - for every phenomenon that the system handles. Thus, measuring the coverage of annotated phenomena by rule applications is useful for discovering to what extent rule bases participate in the recognition of entailment at places where the phenomena exist according to human judgments.

[6] Additional world knowledge is required to infer that *found with his throat cut* entails *died*; given that, the entailment can be validated.

Table 2. Counters of annotations in RTE 1-3 separated into development and test sets. $A_{\#}$ indicates the number of annotations, $P_{\#}$ indicates the number of entailment pairs containing an annotation and $P_{\%}$ indicates the portion of annotated pairs relative to the total amount of entailment pairs.

Ann.	RTE 1						RTE 2						RTE 3					
	Dev set			Test set			Dev set			Test set			Dev set			Test set		
	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$	$A_{\#}$	$P_{\#}$	$P_{\%}$
APP	97	87	31	161	134	34	178	149	37	155	135	34	162	128	31	166	136	33
CONJ	90	79	28	126	112	28	140	118	30	161	144	36	157	121	29	162	134	33
RMOD	180	124	44	243	167	42	311	204	51	394	236	59	262	176	43	306	193	47
Any	367	210	74	530	297	74	629	316	79	710	350	88	581	293	71	635	328	80

Table 3. Results of Consistency Checks - In Check 1, 50 entailment pairs were analyzed and 66% of the annotations that were marked were identical; in Check 2, 70 pairs were analyzed and 74.11% of the annotations that were marked were identical. On average, 70.73% of the annotations we checked were identical. The rubric *Ambiguities* presents cases of structural or modifier-attachment ambiguity in the text that led to divergent annotations. *Major mistakes* are cases of missing annotations, and *Minor mistakes* are cases of divergent annotations (not stemming from ambiguity) or incorrect scope.

Measure	Check 1	Check 2
Entailment Pairs	50	70
Sources	RTE 1	RTE 1 + 2
Total Annotations	93	112
Identical Annotations	62	83
Ambiguities	9	18
Major mistakes	2	7
Minor mistakes	10	4
Consistency (%)	66.67	74.11

Table 4. Sample of Syntactic Rules in BIUTEE

Type	Description
Passive-Active	Transform "X Verb$_{active}$ Y" to"Y is Verb$_{passive}$ by X"
Apposition	Extract an NP and its apposition to an independent IS-A construction
Genitive	Substitute an "X's Y" construction with a "the Y of X" construction

Table 5. Accuracy of BIUTEE in two configurations tested on RTE 1-3

Config	RTE 1	RTE 2	RTE 3
C1	55.38%	61.38%	66.13%
C2	55.88%	61.75%	65.75%

- Text:

```
<Annotation Id=“1833” Type=“conjunction” StartNode=“11404”
EndNode=“11453”>
<Feature><Name>E1</Name><Value>Shapour Bakhtiar</Value>
</Feature>
<Feature><Name>E2</Name><Value>his secretary Sorush Katibeh</Value>
</Feature>
<Feature><Name>construction_id</Name><Value>2</Value></Feature>
</Annotation>
```

- Hypothesis:

```
<Annotation Id=“1834” Type=“reference_to” StartNode=“11511”
EndNode=“11527”>
<Feature><Name >construction_id</Name><Value >2</Value></Feature>
</Annotation>
```

Fig. 2. Annotations tags of *conjunction* in the Text and Hypothesis

Table 6. Accuracy of top three recognizers that participated in the RTE 1-3 challenges. The data presented in this table is based on the results reported in [19].

RTE 1		RTE 2		RTE 3	
System	Accuracy	System	Accuracy	System	Accuracy
Autonma de Madrid [26]	70%	LCC [16]	75.38%	LCC [17]	80%
Ca Foscari Venice [11]	60.6%	LCC [31]	73.75%	LCC [32]	72.25%
MITRE [4]	58.6 %	DISCo [33]	63.88%	“Al. I. Cuza” [18]	69.13%

4.1 Using the Annotations

Measuring the coverage of annotated phenomena by the rule base of an entailment recognizer is done in five steps:

1. Mapping between the rule base and the annotation scheme. This is done manually by pairing each rule with an annotation that corresponds to the same inferential phenomenon.[7]
2. Extracting data from the annotations by parsing the annotated RTE XML file. This includes information on the span of text marked in each annotation and the type of annotation (e.g. *conjunction*).
3. Extracting data from the log file of the recognizer which indicates the rules that the recognizer applied in processing the RTE corpus.
4. Counting the entailment pairs that include an annotation whose span of text was processed by the recognizer using a rule corresponding to the same phenomenon (according to the mapping done in step 1).
5. Dividing the number obtained in step 4 by the total number of entailment pairs in the corpus.

[7] Rules that correspond to unannotated phenomena are ignored.

Section 4.3 describes an execution of this procedure on the Bar Ilan University Textual Entailment Engine (BIUTEE) [30].

4.2 The BIUTEE System

Given a text T and a hypothesis H, BIUTEE is designed to seek for a proof - a sequence of textual inferential steps applied on T that turns T into a text that equals or contains H. Following the design first introduced in [3], BIUTEE employs a syntactic rule base comprised of hand-crafted rules [22] that represent syntactic transformations. The rules in the system are defined based on dependency structures [23]; in Table 4 several of these rules are illustrated in text.

Ideally, a sequence of such transformations should be able to turn T into H. In practice, however, the rule coverage is limited, in that many inferential paradigms that appear in the RTE dataset are not covered by the rule base. Consequently, in addition to the rule base, BIUTEE uses a set of on-the-fly operations aimed at maximizing the similarity between T and H by manipulating T through adding / moving / replacing words.[8]. Furthermore, BIUTEE utilizes lexical and lexical-syntactic rules learned from knowledge resources such as, inter alia, WordNet [12,25], FrameNet [1], VerbOcean [7], Catvar [15], Lin Similarity [20], and DIRT [21].

In this architecture, the system always reaches a proof from T to H. In order to identify proofs that capture valid entailments, the system uses a confidence model that assigns a confidence value to each rule and on-the-fly operations that are performed as part of a proof. If the computed total confidence value of a proof is high enough, it is assumed to represent a valid entailment, and vice versa. The function that assigns confidence values to rules and on-the-fly operations, and the threshold for predicting a valid/invalid entailment are learned automatically based on the training set (see [30] for full details).

4.3 Examination of BIUTEE's Rule Base

We executed the procedure described in 4.1 on BIUTEE. Information on all instances of rule application was extracted from the log file of the system, and all the rules that had been applied were either mapped to annotation labels or classified as relevant to unannotated (ignored) phenomena. We ran the system in two comparable configurations:

- C1 - Basic configuration with only syntactic rules included.
- C2 - Resource-based configuration with lexical and lexical/syntactic rules based on WordNet, FrameNet, VerbOcean, Catvar and DIRT, as well as the syntactic rules.

[8] The goal of the transformations BIUTEE applies to the graph that represents T's dependency structure is to turn it into a graph that contains H as a subgraph.

Table 5 displays the performance of the system in configurations C1 and C2. Table 6 reports the performance of the top three recognizers that participated in the RTE 1-3 challenges. In Table 7 we present our analysis of the coverage of annotated phenomena by the rule base.[9]

Table 7. Coverage of annotated phenomena by BIUTEE's rule base in two configurations tested on RTE 1-3. P_{Ann} stands for pairs containing an annotation of APP or CONJ, and $P_{Rule-Cx}$ stands for pairs that BIUTEE, running in configuration *Cx*, processed by applying a rule of APP or CONJ on an annotated phenomenon. $P_{\#}$ indicates the number of entailment pairs in each set (P_{Ann} or $P_{Rule-Cx}$ respectively) and $P_{\%}$ indicates the portion of a set in the total amount of entailment pairs (in percents). On average, BIUTEE processes only 4.52% of the entailment cases based on rules that correspond to the phenomena annotated in the corpus.

Pairs set	RTE 1		RTE 2		RTE 3	
	$\mathrm{P}_{\#}$	$\mathrm{P}_{\%}$	$\mathrm{P}_{\#}$	$\mathrm{P}_{\%}$	$\mathrm{P}_{\#}$	$\mathrm{P}_{\%}$
P_{Ann}	210	53	241	60	239	58
$\mathrm{P}_{Rule-C1}$	2	1	37	9	3	1
$\mathrm{P}_{Rule-C2}$	4	1	47	12	15	4

4.4 Summary of Results

In both configurations, our results indicate that the coverage of annotated phenomena by BIUTEE's rule base is rather low: 1% on RTE 1, 9-12% on RTE 2 and 1-4% on RTE 3. This means that the rule base is only rarely used for processing the entailment patterns annotated in our corpus. In practice, in order to process these linguistic phenomena and to predict the entailment, the system uses mostly on-the-fly operations.[10]

5 Conclusions

We have presented a semantic annotation of inferential phenomena in the RTE corpus for the purpose of evaluating and boosting RTE systems. We have chosen to focus on the annotation of semantic phenomena which are predominant in the RTE and can be annotated with high consistency, but which may have several syntactic expressions and therefore allow us to generalize regarding abstract entailment patterns. A use case of the corpus is presented, based on an examination of the rule base employed by the BIUTEE recognizer. This study found

[9] BIUTEE handles constructions that manifest RMOD internally, without rule application. The system does not report on instances of this phenomenon, and due to that we do not include RMOD in this examination.

[10] The marginal effect of the knowledge resources used in C2 on the overall accuracy of BIUTEE is in line with the results of ablation studies that were performed on various RTE systems (see [6] for more information).

that the rule base was rarely applied to the targeted phenomena. While we did not analyze the implications of these results for the architecture of BIUTEE, we believe that they show that the annotated corpus is useful for further research on entailment phenomena that are treated by rule bases, as in BIUTEE and other RTE systems (e.g. [17] and [18]). In addition, we think that the annotated data is also a useful benchmark for automatic learning of inferential phenomena recognition and processing.

Acknowledgment. The work of Assaf Toledo, Sophia Katrenko, Stavroula Alexandropoulou, Heidi Klockmann and Yoad Winter was supported by a VICI grant number 277-80-002 by the Netherlands Organisation for Scientific Research (NWO). The work of Asher Stern and Ido Dagan was partially supported by the Israel Science Foundation grant 1112/08 and the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287923 (EXCITEMENT).

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley FrameNet project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, ACL 1998, vol. 1, pp. 86–90. Association for Computational Linguistics, Stroudsburg (1998)
2. Bar Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The second pascal recognising textual entailment challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (2006)
3. Bar-Haim, R., Dagan, I., Greental, I., Szpektor, I., Friedman, M.: Semantic inference at the lexical-syntactic level for textual entailment recognition. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 131–136 (2007)
4. Bayer, S., Burger, J., Ferro, L., Henderson, J., Yeh, E.: Mitres submission to the eu pascal rte challenge. In: Proc. of the First PASCAL Challenge Workshop on Recognizing Textual Entailment, pp. 41–44 (2005)
5. Bentivogli, L., Cabrio, E., Dagan, I., Giampiccolo, D., Leggio, M.L., Magnini, B.: Building textual entailment specialized data sets: a methodology for isolating linguistic phenomena relevant to inference. In: Proceedings of LREC 2010 (2010)
6. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The fifth pascal recognizing textual entailment challenge. Proceedings of TAC 9, 14–24 (2009)
7. Chklovski, T., Pantel, P.: Verbocean: Mining the web for fine-grained semantic verb relations. In: Proceedings of EMNLP, vol. 4, pp. 33–40 (2004)
8. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE, Version 6 (2011)
9. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 177–190. Springer, Heidelberg (2006)

10. Del Gobbo, F.: On the syntax and semantics of appositive relative clauses. In: Dehe, N., Kavalova, Y. (eds.) Parentheticals, pp. 173–201 (2007)
11. Delmonte, R., Tonelli, S., Boniforti, M.A.P., Bristot, A., Pianta, E.: VENSES – A Linguistically-Based System for Semantic Evaluation. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 344–371. Springer, Heidelberg (2006)
12. Fellbaum, C.: WordNet: An Electronic Lexical Database. Language, Speech, and Communication. MIT Press (1998)
13. Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E.: The fourth pascal recognising textual entailment challenge. In: TAC 2008 Proceedings (2008)
14. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third pascal recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, RTE 2007, pp. 1–9. Association for Computational Linguistics, Stroudsburg (2007)
15. Habash, N., Dorr, B.: A categorial variation database for english. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 17–23 (2003)
16. Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., Shi, Y.: Recognizing textual entailment with LCCs GROUNDHOG system. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (2006)
17. Hickl, A., Bensley, J.: A discourse commitment-based framework for recognizing textual entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 171–176 (2007)
18. Iftene, A., Balahur-Dobrescu, A.: Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 125–130 (2007)
19. Iftene, A.: Textual Entailment. Ph.D. thesis, ”Al. I. Cuza” University, Iasi, Romania (March 2009)
20. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 17th International Conference on Computational Linguistics, vol. 2, pp. 768–774 (1998)
21. Lin, D., Pantel, P.: DIRT@ SBT@ discovery of inference rules from text. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 323–328 (2001)
22. Lotan, A.: A Syntax-based Rule-base for Textual Entailment and a Semantic Truth Value Annotator. MA thesis, Department of Linguistics, Tel Aviv University (2012)
23. de Marneffe, M.C., Manning, C.D.: The stanford typed dependencies representation. In: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, Coling 2008, pp. 1–8 (2008)
24. McCawley, J.D.: Parentheticals and discontinuous constituent structure. Linguistic Inquiry 13(1), 91–106 (1982)
25. Miller, G.A.: WordNet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
26. Perez, D., Alfonseca, E.: Application of the bleu algorithm for recognising textual entailments. In: Proceedings of the First Challenge Workshop Recognising Textual Entailment, pp. 9–12 (2005)
27. Quirk, R., Greenbaum, S., Leech, G., Svartvik, J.: A Comprehensive grammar of the English language. Longman, London (1985)

28. Sammons, M., Vydiswaran, V.G., Roth, D.: Ask not what textual entailment can do for you... In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1199–1208 (2010)
29. Sells, P.: Restrictive and non-restrictive modification. CLSI Report, Center for the Study of Language and Information, vol. (28). Stanford University (1985)
30. Stern, A., Dagan, I.: A confidence model for syntactically-motivated entailment proofs. In: Proceedings of RANLP 2011 (2011)
31. Tatu, M., Iles, B., Slavick, J., Novischi, A., Moldovan, D.: Cogex at the second recognizing textual entailment challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 104–109 (2006)
32. Tatu, M., Moldovan, D.: COGEX at RTE3. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 22–27 (2007)
33. Zanzotto, F.M., Moschitti, A., Pennacchiotti, M., Pazienza, M.T.: Learning textual entailment from examples. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment (2006)

Recognizing Textual Entailment in Non-english Text via Automatic Translation into English

Partha Pakray[1], Snehasis Neogi[1], Sivaji Bandyopadhyay[1], and Alexander Gelbukh[2]

[1] Computer Science and Engineering Department,
Jadavpur University, Kolkata, India
[2] Center for Computing Research, National Polytechnic Institute,
Mexico City, Mexico
{parthapakray,snehasis1981}@gmail.com,
sbandyopadhyay@cse.jdvu.ac.in
www.gelbukh.com

Abstract. We show that a task that typically involves rather deep semantic processing of text—being recognizing textual entailment our case study—can be successfully solved without any tools at all specific for the language of the texts on which the task is performed. Instead, we automatically translate the text into English using a standard machine translation system, and then perform all linguistic processing, including syntactic and semantic levels, using only English language linguistic tools. In this case study we use Italian annotated data. Textual entailment is a relation between two texts. To detect it, we use various measures, which allow us to make entailment decision in the two-way classification task (YES / NO). We set up various heuristics and measures for evaluating the entailment between two texts based on lexical relations. To make entailment judgments, the system applies named entity recognition module, chunking, part-of-speech tagging, *n*-grams, and text similarity modules to both texts, all those modules being for English and not for Italian. Rules have been developed to perform the two-way entailment classification. Our system makes entailment judgments basing on the entailment scores for the text pairs. The system was evaluated on Italian textual entailment data sets: we trained our system on Italian development datasets using the WEKA machine learning toolset and tested it on Italian test data sets. The accuracy of our system on the development corpus is 0.525 and on the test corpus is 0.66, which is a good result given that no Italian-specific linguistic information was used.

Keywords: Recognizing textual entailment, *n*-grams, text similarity, machine translation, cross-lingual textual entailment.

1 Introduction

Recognizing Textual Entailment (RTE) is one of recent challenges of Natural Language Processing (NLP) [1]. Textual entailment is defined as a directional relationship between pairs of text expressions, denoted by T—the entailing "Text", and H—the entailed "Hypothesis". T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 26–35, 2013.

For example, the following text: T = *John's assassin is in jail* entails the following hypothesis: H = *John is dead*. Indeed, if there exists one's assassin, then this person is dead. On the other hand, the text T = *Mary lives in Europe* does not entail a hypothesis H = *Mary lives in London*.

RTE is useful for many NLP tasks. For example, in text summarization (sometimes denoted by SUM) a summary of a document should be entailed by its contents; paraphrases can be seen as mutual entailment between the two expressions; in Information Extraction (IE), the extracted information should also be entailed by the documents; in Question Answering (QA), the answer obtained for a question must be entailed by the supporting snippets of text.

There exist a number of Recognizing Textual Entailment evaluation initiatives. There have been held seven Recognizing Textual Entailment (RTE) competitions: RTE-1 in 2005 [2], RTE-2 [3] in 2006, RTE-3 [4] in 2007, RTE-4 [5] in 2008, RTE-5 [6] in 2009, RTE-6 [7] in 2010, RTE-7 [8] in 2011. In 2010, Parser Training and Evaluation using Textual Entailment event [9] was organized in frame of SemEval-2.

In 2011, Recognizing Inference in Text (RITE) was organized by NTCIR-9.[1] In 2012, Cross-lingual Textual Entailment for Content Synchronization (CLTE)[2] track was organized in frame of SemEval-2012. Gradual advances and previous versions of the present work have been presented at RTE-5, RTE-6, RTE-7, SemEval-2 Parser Training and Evaluation using Textual Entailment Task, RITE, and SemEval-2012: Cross-lingual Textual Entailment for Content Synchronization.

In this paper, we report the results obtained with an improved version of our system. This system uses a chain of NLP modules in order to obtain a wide variety of features of both text T and hypothesis H, varying from *n*-gram based to syntactic and semantic levels.

The contribution of this paper consists in suggesting that certain tasks—in this case the recognizing textual entailment task as a case study—that involve deep language processing can be accomplished without the use of any tools or techniques specific for the given language.

Namely, we use a pivot language approach: our text processing modules work with English language data; the input texts in any language are translated into English using a standalone machine translation system. Thus, we show that machine translation can be used to successfully perform the RTE task in any language or even when T and H are in different languages.

For evaluation, in this work we use the EVALITA Textual Entailment data sets. EVALITA 2009[3] was an evaluation campaign of both Natural Language Processing and speech technologies for Italian language. The EVALITA Textual Entailment task[4] consisted in detection of inferential relationships between pairs of short texts.

The work is organized as follows. Our two-way textual entailment recognition system architecture is presented in Section 2. Section 3 describes feature extraction, to be used with the WEKA toolset [10]. The experimental results on the development and test data sets are given in Section 4. Finally, conclusions are drawn in Section 5.

[1] `http://artigas.lti.cs.cmu.edu/rite/Main_Page`

[2] `http://www.cs.york.ac.uk/semeval-2012/task8/`

[3] `http://www.evalita.it/2009`

[4] `http://www.evalita.it/2009/tasks/te`

2 System Architecture: A Machine Learning Approach

The EVALITA datasets used to test our system are available in the Italian language. The task that has been proposed for EVALITA is basically a binary classification textual entailment problem: a system should predict whether the text T in the text pair entails or not the corresponding hypothesis H.

We explore in this paper a machine learning based approach for this EVALITA task. Our system generates various lexical matching scores calculated over the development dataset are used to train the model along with the target class. Specifically, the system includes such components as the preprocessing module, lexical similarity module, and text similarity module. The lexical similarity module is in turn divided into sub-modules such as POS matching, chunk matching, and named entity matching.

This trained model was then used to predict the classification of unseen text pairs in the test dataset. The WEKA machine learning toolset is used to classify and predict the classification of text pairs. As the pairs are available in Italian language, our system uses pivot language approach: it applies the textual entailment module after automatically translating the text pair into English.

Figure 1 shows our system architecture, where the text and hypothesis sentences are translated into English.

2.1 Pre-processing

The system extracts the T (text) and H (hypothesis) pair from the EVALITA task data. The text and hypothesis pair is available there in the Italian language. Microsoft Bing translator API[5] for Bing translator (the file `microsoft-translator-java-api-0.4-jar-with-dependencies.jar`) was used to translate the T and H text sentences into English.

The translated Text and Hypothesis sentences were then passed through stop-word removal and co-reference resolution modules.

Stop-Word Removal. This module removes stop-words listed in a pre-defined stop-word list from the T and H sentences.

Co-reference Resolution. Co–reference chains are evaluated in the datasets before passing the text to the RTE module. The objective is to increase the entailment score by substituting the anaphors with their antecedents.

A word (often a pronoun) or phrase in the sentence can be used to refer to an entity introduced earlier or later in the discourse. The description that introduces the entity and all expressions that refer to it are said to have the same referent; this phenomenon is called co-reference.

We distinguish between two types of co-reference. When the reader must look back to the previous context to find the referent, then the co-reference is called

[5] `http://code.google.com/p/microsoft-translator-java-api/`

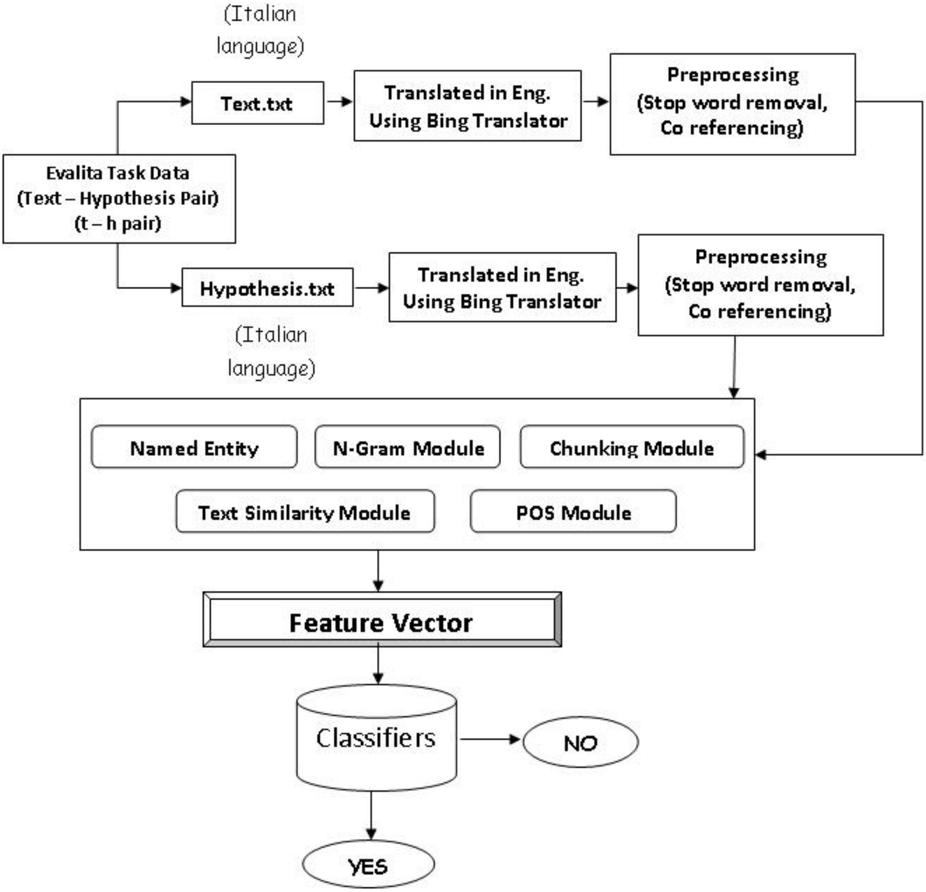

(Italian language)
Text.txt
Translated in Eng. Using Bing Translator
Preprocessing (Stop word removal, Co referencing)
Evalita Task Data (Text – Hypothesis Pair) (t – h pair)
Hypothesis.txt
Translated in Eng. Using Bing Translator
Preprocessing (Stop word removal, Co referencing)
(Italian language)
Named Entity
N-Gram Module
Chunking Module
Text Similarity Module
POS Module
Feature Vector
Classifiers
NO
YES

Fig. 1. Architecture of our system

anaphoric reference. When the reader must look forward to find the referent, then it is called cataphoric reference.

To address this problem we used a tool called JavaRAP[6], which is a Java-based implementation of the Resolution of Anaphora Procedure (RAP), an algorithm developed by Lappin and Leass [11]. We observed, however, that co-referential expressions are very rare in the sentence based paradigm.

2.2 Lexical Based Textual Entailment (TE) Recognition Module

Text–Hypothesis pairs are the inputs to the system. The overall TE module is a collection of several lexical-based sub-modules. Each sub-module produces a lexical matching score that is used to develop a training model.

[6] http://aye.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.html

N-gram Matching Module. The *n*-gram matching basically measures the percentage of unigrams, bigrams, and trigrams of the Hypothesis that are also present in the corresponding Text. Note that both the Text and the Hypothesis have been previously pre-processed with stop-word removal, so that the *n*-grams do not contain any stop-words.

The scores for different values of *n* are simply combined to get an overall *n*-gram matching score for a particular pair.

Chunk Similarity Module. This sub-module of our system evaluates the key NP chunks of both text and hypothesis, that are identified using the NP Chunker v1.1[7] tool. Then, our system checks the presence of NP chunks of the hypothesis in the corresponding text.

The system calculates the overall value for the chunk matching, i.e., the number of NP chunks of the text that match with NP chunks of the hypothesis. If the chunks are not similar in their surface form, then our system applies WordNet-based matching for the words: if they match in WordNet synsets information, then the chunks are considered similar.

WordNet is one of most important resource for lexical analysis. The WordNet 2.0 has been used for WordNet-based chunk matching. The API for WordNet Searching (JAWS),[8] an API that provides Java applications with the ability to retrieve data from the WordNet database, was used.

Text Distance Module. The system takes into consideration a wide variety of text similarity measures calculated over the each T–H pair. These text similarity measures are summed up together to produce the total final score for a particular text pair, which is used for the classification decision.

Specifically, the following well-known text similarity measures are used in our system:

- Cosine Similarity
- Levenshtein Distance
- Euclidean Distance
- Monge–Elkan Distance
- Needleman–Wunch Distance
- Smith–Waterman Distance
- Block Distance
- Jaro Similarity
- Matching Coefficient Similarity
- Dice Similarity
- Overlap Coefficient
- Q-grams Distance

[7] http://www.dcs.shef.ac.uk/~mark/phd/software/

[8] http://lyle.smu.edu/~tspell/jaws/index.html

Named Entity Matching. This process is based on the detection and matching of named entities (NEs) in the T–H pair. The Stanford Named Entity Recognizer (NER) was used to tag the named entities in both text and hypothesis. The system simply calculates the number of the Hypothesis's NEs that are present in the Text. A score is associated with the matching as follows:

$$\text{NE\_Match} = \frac{\text{Number of common NEs in Text and Hypothesis}}{\text{Number of NEs in Hypothesis}}\,.$$

Part-of-Speech Matching. This module basically deals with matching the common part of speech (POS) tags between the T and H. The Stanford POS tagger was used to tag words with the parts of speech in both the Text and the Hypothesis. The system calculates the number of the verb and noun POS words in the Hypothesis that match those in the Text. A score is associated with the number of patched POSs as follows:

$$\text{POS\_Match} = \frac{\text{Number of verb and noun POSs in Text and Hypothesis}}{\text{Total number of verb and noun POSs in Hypothesis}}\,.$$

3 Feature Extraction

The system-generated matching scores were fed in the training module of the WEKA machine learning tool to develop a classification model. This model is used to predict the presence or absence of entailment in the untagged text pair in the test set of the EVALITA task.

The main motivation to introduce a machine learning approach in this EVALITA task is that the Textual Entailment task can be considered as a classification problem. Different measures applied to the Text–Hypothesis pair can be used as a feature vector for the classifier. In this architecture we used lexical similarities as the feature vector. Naïve Bayes classifier was used to predict the outcome for unseen pairs. Figure 2 illustrates the concept of machine learning for the classification of textual entailment problem.

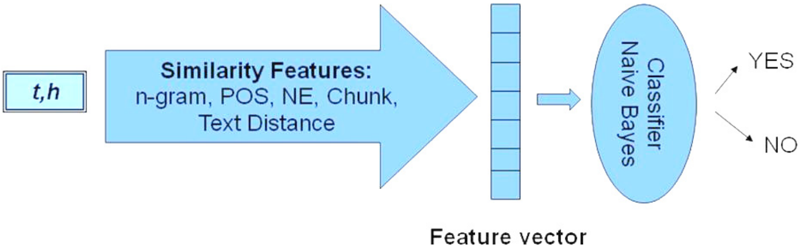


Fig. 2. Machine Learning Classification

The Naïve Bayes algorithm is a classification algorithm based on the Bayes rule that assumes the attributes $X_1, \ldots, X_n$ are all mutually independent, given a condition Y. The importance of this assumption lays in that it dramatically simplifies the representation of the conditional probability P(X|Y), as well as the problem of estimating it from the training data.

Consider, for example, the case of two variables, where the feature vector X = <X_1, X_2>. In this case we have:

$$\begin{aligned} P(X \mid Y) &= P(X_1, X_2 \mid Y) \\ &= P(X_1 \mid X_2, Y)\, P(X_2 \mid Y) \\ &= P(X_1 \mid Y)\, P(X_2 \mid Y), \end{aligned}$$

where the second line follows from a general property of probabilities, and the third line follows from the definition of conditional independence. More generally, when the feature vector X contains *n* attributes that are mutually independent given Y, we have:

$$P(X_1,\dots,X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y) \tag{1}$$

For the sake of completeness, let us now derive the Naive Bayes algorithm, assuming in general that Y is any discrete-valued variable, and the attributes X_1, …, X_n are any discrete or real valued attributes. Our goal is to train a classifier that will output the probability distribution over possible values of the target Y, for each new instance X that is the data point to be classified.

The expression for the probability that Y will take on its *k*-th possible value, according to the Bayes rule, is

$$P(Y = y_k \mid X_1,\dots,X_n) = \frac{P(Y = y_k)P(X_1,\dots,X_n \mid Y = y_k)}{\sum_j P(Y = y_j)P(X_1,\dots,X_n \mid Y = y_j)}, \tag{2}$$

where the sum is taken over all possible values y_j of Y. Now, assuming that all X_i are conditionally independent given Y, we can rewrite the above equation as

$$P(Y = y_k \mid X_1,\dots,X_n) = \frac{P(Y = y_k)\prod_i P(X_i \mid Y = y_k)}{\sum_j P(Y = y_j)\prod_i P(X_i \mid Y = y_j)}. \tag{3}$$

This is the fundamental equation underlying the Naïve Bayes classifier (called naïve because the independent assumption if often assumed without thorough justification).

The training file comprises different lexical similarity matching scores, separated by comma. It also includes the target class of each text pair from the gold standard values. An example of the training file in the WEKA format is shown in Figure 3 (obviously, not all data rows are shown).

This file is fed into the WEKA toolset, with the Naïve Bayes option for the classification algorithm to use. The toolset automatically evaluates the average accuracy of the classification achieved on these training data.

```
@relation EVALITA
@attribute N-Gram real
@attribute Text-Similarity real
@attribute Part-of-Speech real
@attribute Named Entity real
@attribute Chunk real
@attribute class {YES,NO}
@data
24,16,10,2,15,YES
39,12,23,0,17,YES
41,15,17,1,11,YES
61,13,28,3,21,YES
78,16,34,0,9,NO
```

Fig. 3. Example of the feature vector structure of the training data

4 Experimental Results

The development and test datasets consist of 400 Text–Hypothesis pairs. The lexical features are calculated for both development and test datasets. Matching scores of the development dataset were used to train the model. The WEKA toolset was used to train the classification model and test the output for the unseen pairs thereafter.

The experimental results obtained for both development and text data predicted by the WEKA toolset using the Naïve Bayes as the classification algorithm are as follows.

The confusion matrix for the development data is shown in Table 1.

Table 1. Confusion matrix obtained on the training dataset

Correctly Classified Instances	210	52.5%
Incorrectly Classified Instances	190	47.5%
Total Number of Instances	400	

The precision, recall and the corresponding F-measure for the development dataset are shown in Table 2.

Table 2. Precision, recall, an F-measure obtained on the training dataset.

Class	Precision	Recall	F-measure
YES	0.541	0.905	0.677
NO	0.344	0.061	0.104
Weighted Avg.	0.452	0.525	0.419

The accuracy for the test dataset was 0.525.

The confusion matrix for test data is shown in Table 3.

Table 3. Confusion matrix obtained on the test dataset

Correctly Classified Instances	264	66.0 %
Incorrectly Classified Instances	136	34.0%
Total Number of Instances	400	

The precision, recall and the corresponding F-measure obtained on the test dataset are shown in Table 4.

Table 4. Precision, Recall, F-Measure on Test Data

Class	Precision	Recall	F-Measure
YES	0.602	0.945	0.735
NO	0.872	0.375	0.524
Weighted Avg.	0.737	0.660	0.630

Finally, the accuracy obtained for test dataset was 0.660. This is a very good accuracy given that no language-specific (for the Italian language) tools were used for feature extraction. Instead, all linguistic processing was performed on the English text obtained via automatic translation.

5 Conclusions and Future Work

Results show that a lexical based approach appropriately tackles the textual entailment problem. Experiments have been initiated for a semantic and syntactic based RTE task.

The next step is to carry out detailed error analysis of the present system and identify ways to overcome the errors. In the present task, the final textual entailment system has been optimized for the entailment YES/NO decision using the development set.

Finally, our textual entailment system is to be applied in Japanese, French, Italian, Spanish, and German datasets also. With those experiments we expect to show that the idea of using only English-language linguistic information for deep processing of data in other languages can be applied to a wide variety of languages, most probably depending on the quality of the automatic translation system available for this. We also plan to investigate the applicability of this idea to the cross-lingual or multilingual settings: when the hypothesis and the text are in different languages, and the training and test datasets contain pairs in different combinations of languages.

Acknowledgements. The work was partially supported by the Governments of India and Mexico under the CONACYT-DST India (CONACYT 122030) project "Answer Validation through Textual Entailment", the Government of Mexico under the CONACYT 50206-H project and SIP-IPN 20121823 project through Instituto Politécnico Nacional, and the Seventh Framework Programme of European Union, project 269180 "Web Information Quality Evaluation Initiative (WIQ-EI)".

References

1. Ledeneva, Y., Sidorov, G.: Recent Advances in Computational Linguistics. Informatica. International Journal of Computing and Informatics 34, 3–18 (2010)
2. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First PASCAL Recognizing Textual Entailment Workshop (2005)
3. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006)
4. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic (2007)
5. Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E.: The Fourth PASCAL Recognizing Textual Entailment Challenge. In: Text Analysis Conference (TAC) 2008 Notebook Proceedings (2008)
6. Bentivogli, L., Dagan, I., Dang, H.T., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the Text Analysis Conference (TAC) 2009 Workshop. National Institute of Standards and Technology, Gaithersburg (2009)
7. Bentivogli, L., Clark, P., Dagan, I., Dang, H.T., Giampiccolo, D.: The Sixth PASCAL Recognizing Textual Entailment Challenge. In: Text Analysis Conference (TAC) 2010 Notebook Proceedings (2010)
8. Bentivogli, L., Clark, P., Dagan, I., Dang, H., Giampiccolo, D.: The Seventh PASCAL Recognizing Textual Entailment Challenge. In: Text Analysis Conference (TAC) 2011 Notebook Proceedings (2011)
9. Yuret, D., Han, A., Turgut, Z.: SemEval-2010 Task 12: Parser Evaluation using Textual Entailments. In: Proceedings of the SemEval 2010 Evaluation Exercises on Semantic Evaluation (2010)
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)
11. Lappin, S., Leass, H.: An Algorithm for Pronominal Anaphora Resolution. Computational Linguistics 20(4), 535–561 (1994)

SMSFR: SMS-Based FAQ Retrieval System

Partha Pakray[1], Santanu Pal[1], Soujanya Poria[1], Sivaji Bandyopadhyay[1],
and Alexander Gelbukh[2]

[1] Computer Science and Engineering Department,
Jadavpur University, Kolkata, India
[2] Center for Computing Research,
National Polytechnic Institute, Mexico City, Mexico
{parthapakray,santanupersonal1,soujanya.poria}@gmail.com,
sbandyopadhyay@cse.jdvu.ac.in
www.gelbukh.com

Abstract. The paper describes an SMS-based FAQ retrieval system. The goal of this task is to find a question Q* from corpora of FAQs (Frequently Asked Questions) that best answers or matches the SMS query S. The test corpus used in this paper contained FAQs in three languages: English, Hindi and Malayalam. The FAQs were from several domains, including railway enquiry, telecom, health and banking. We first checked the SMS using the Bing spell-checker. Then we used the unigram matching, bigram matching, and 1-skip bigram matching modules for monolingual FAQ retrieval. For cross-lingual system, we used the following three modules: an SMS-to-English query translation system, an English-to-Hindi translation system, and cross-lingual FAQ retrieval.

Keywords: domain-specific information retrieval, *n*-grams, spell checker, Bing translator.

1 Introduction

In spite of great advances of information retrieval systems and associated natural language processing technologies [1], domain-specific retrieval systems and retrieval systems used with special types of queries continue to represent a challenge for current technology and to be a topic of active research.

The Forum for Information Retrieval Evaluation (FIRE)[1] [2] is a forum for Information Retrieval evaluation that is traditionally mainly focused on Indian languages. After the success of FIRE 2008 and FIRE 2010, the main goal of the FIRE 2011 was to support the continuity of research on Information Retrieval, with the stress on non-English languages, specifically those used on the Indian sub-continent.

India, with its huge population, has a very high rate of using mobile phones, with service costs low enough for even very poor people to use the phones actively. Accordingly, FIRE 2011 included an SMS-based FAQ retrieval [3] task. The goal of this task was to find a question Q* from corpora of FAQs (Frequently Asked Questions) that best answers or matches a given SMS query S.

[1] http://www.isical.ac.in/~clia/

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 36–45, 2013.

SMS queries, which are written in “SMS language” specific for this kind of communication and different from the usual grammatically correct text, tend to be noisy, because the users try and compress text by omitting letters, by using slang, etc., due to a restriction on the length of such messages (at most 160 characters are allowed for one SMS), the lack of screen space, which makes reading large amounts of text difficult, and other similar reasons.

The messages also frequently contain unintended typographical errors due to small size of keypads on mobile phones as also poor language skills of the users (which, in turn, are caused both by the availability of mobile services to the poorest strata of the population with low literacy and the fact that major languages used in India are not native languages for a large share of the population). The presence of such noise makes this task different and more challenging than traditional Question Answering (QA) retrieval tasks.

The SMS retrieval task consists of three sub-tasks.

Task 1: Monolingual FAQ Retrieval: In this sub-task, the SMS query and the FAQ corpus is in the same language. The goal in this task is to find the best matching question Q* from a monolingual collection of FAQs Q that matches an SMS query expressed in the same language as the FAQs. This task is most similar to the classical information retrieval task.

Task 2: Cross-lingual FAQ Retrieval: In this sub-task, the SMS query and the FAQ corpus are in different languages. That is, if the SMS query is in Language L1 (for example, Malayalam), then the FAQ corpus will be in a language L2 other than L1, for example, in English. Thus, the goal in this task is to find the best matching question Q* from the set of FAQs in language L2 while the SMS query is in a different language L1.

Task 3: Multi-lingual FAQ Retrieval: In this sub-task the SMS queries can be in multiple languages and these can match to FAQ collections in multiple languages. For example, SMS queries could be written in English, or in Hindi, or in Malayalam, and these queries could match FAQ collections of FAQs in all languages that participate in the task. That is, the goal is to find a pertinent FAQ item that can be in English, Hindi, or Malayalam.

In this paper we report the systems developed for the monolingual task, Task 1, and for the cross-lingual task, Task 2, while developing a system for Task 3 is the topic of our future work.

The paper is organized as follows. Section 2 briefly presents the datasets used in the work, along with examples and some statistics. Section 3 presents the block diagram of the architecture of our system developed for the monolingual task, Task 1. Section 4 presents the block diagram of the architecture of our system developed for the cross-lingual task, Task 2. Section 5 describes the experimental results obtained with these two systems. Finally, Section 6 concludes the paper and discusses the directions of our future work.

Table 1. Dataset statistics for the SMS-based FAQ retrieval task

FAQ	Language		Number of SMS queries in each task (in-domain / out-of-domain)		
			Monolingual task	Cross-lingual task	Multilingual task
7251	English	Training	701 / 370	291 /181	290 / 170
		Test	728 /2677	37 / 3368	724 / 2681
1994	Hindi	Training	181 / 49		183 / 47
		Test	200 / 124		200 / 124
681	Malayalam	Training	120 / 20		60 / 20
		Test	50 / 0		50 / 0

2 Datasets Used for SMS Based FAQ Retrieval

In the "SMS based FAQ Retrieval" task[2] of FIRE 2011 the datasets used for training and testing of the systems were released. The training FAQ dataset and the training SMS datasets were released in three languages: English, Hindi, and Malayalam. Some statistical data on these datasets are shown in Table 1. In this table, the two figures separated by a slash stand for the number of in-domain items and out-of-domain items in each dataset.

A sample of XML formatted data for the training dataset is shown in Table 2.

3 System Architecture for Task 1: Monolingual

The architecture of the system for the monolingual SMS FAQ retrieval, Task 1, is shown as a block diagram in Figure 1. Our solution for this task is a rule-based system for ranking the candidate FAQ items. Various components of the system are: a preprocessing module, a unigram matching module, a bigram matching module, and the 1-skip bigram matching module.

In the following subsections we give a brief description of the modules used in the system.

3.1 Spellchecking

The SMS and FAQ statements both have various spelling errors. So, to achieve optimized and improved matched between them, both SMS and FAQ statements were passed through the spellchecker module. We used the Bing spellchecker for this purpose, because it is free of charge and has acceptable quality as compared with other available options.

[2] `http://www.isical.ac.in/~fire/faq-retrieval/data.html`

Table 2. Sample XML-formatted training data

FAQ	<FAQ> <FAQID>ENG_CAREER_33</FAQID> <DOMAIN>ENG_CAREER</DOMAIN> <QUESTION>What is an effective resume?</QUESTION> <ANSWER> An effective resume is one which makes your phone ring or your email blink. </ANSWER> </FAQ>
SMS	<SMS> <SMS_QUERY_ID>ENG_5</SMS_QUERY_ID> <SMS_TEXT>are the carier conselling sessionss confidensial</SMS_TEXT> <MATCHES> <ENGLISH>ENG_CAREER_43</ENGLISH> <MALAYALAM>NONE</MALAYALAM> <HINDI>NONE</HINDI> </MATCHES> </SMS>

As Fig. 1 shows, we use the same Bing spellchecker module to process both the SMS dataset and the FAQ dataset, which guarantees that we correctly math coinciding words that are considered ungrammatical and changed by the spellchecker module: while both word are changed by the spellchecker module, they are changed in the same way and thus match normally.

Suppose SMS query is S and FAQ query is F. After spell-checking we obtain the changed form of the query S, which we denote S′, and the changed form of the FAQ statement F, which we denote F′.

3.2 Unigram Matching

After we have obtained the spell-checked statements S′ and F′, we fed them into the unigram matching module.

Suppose the text of the SMS statement S′ contains a word pattern that can be expressed as <L_1, L_2, ..., L_n>, and the text of the FAQ statement F′ contains the word pattern that can be expressed as <W_1, W_2, ..., W_k>. Then we searched for matching of each word of S′ in F′.

If a direct match occurred, then we do not search any further for the word L_i and pick up the next word from the list and search again for that word. We consider that a direct match occurred if there is some FAQ word W_i such that $W_i = L_j$ for some $i \le k$ and $j \le n$.

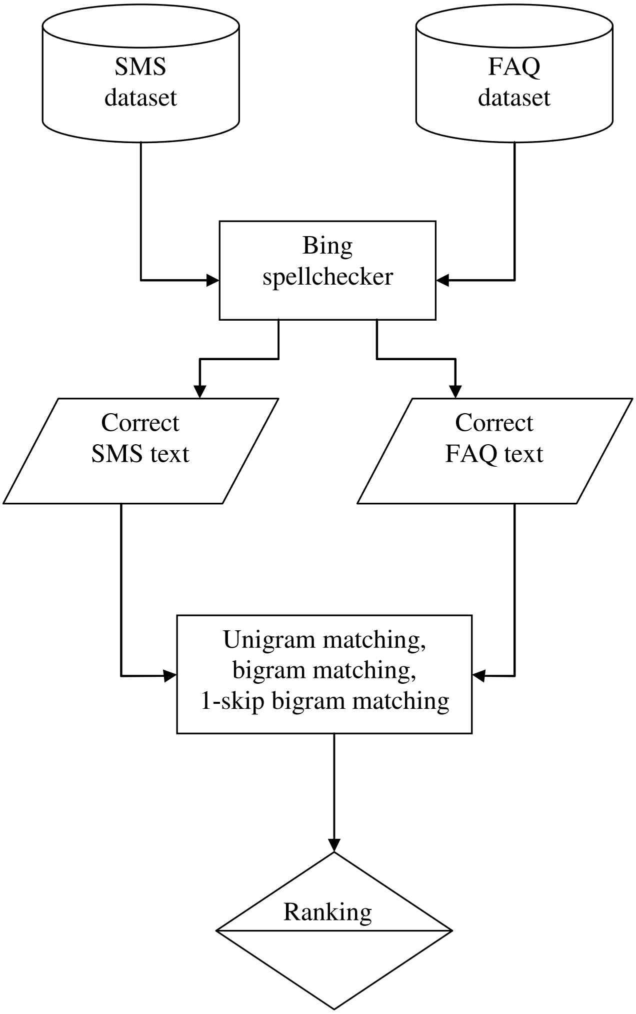


Fig. 1. Monolingual FAQ retrieval system architecture

Now, if there is no direct match, then we looked up the word in WordNet 3.0 [4] and obtained its hyponyms, synonyms, etc., and searched each one of these words in the F′ list. If a match was found, then we passed on to next word.

Otherwise, we searched for an overlap between the synonym and hyponym list of the word L_j and the synonym and hyponym list of the word W_i. If in this case any matching words can be found, then we went on to the next word in the S′ list and store that word as a matched word. Otherwise we skipped the word and proceeded for next word.

The expression used as a score for the unigram matching module was:

$$\text{Unigram Score} = \frac{\text{Number of SMS word that match FAQ words}}{\text{Total number of words in SMS}}. \cdot$$

3.3 Bigram Matching

In this module, we aimed to find a match between two statements by considering the bigram occurrences of their words. We took the two consecutive words from the S′ list, represented as <L_i, L_{i+1}>, and similarly from the F′ list, <W_j, W_{j+1}>. If a match was found, then we went on to the next consecutive bigram.

Otherwise, we looked up in WordNet all hyponyms and synonyms for the words of the bigram <L_i, L_{i+1}>, and similarly we retrieved all hyponyms and synonyms for the words of the bigram <W_j, W_{j+1}>. Suppose we have a synonym list for L_i as (x_1, x_2, ..., x_n) and for L_{i+1} a list (y_1, y_2, …, y_k), and similarly for the pair <W_j, W_{j+1}>, obtaining lists (s_1, s_2, ..., s_n) and (t_1, t_2, …, t_k), correspondingly. If there was any matching words between the lists for the SMS bigram <L_i, L_{i+1}> and the list for the FAQ bigram <W_j, W_{j+1}>, then we considered that a match was found, and proceed to analyze the next bigram sequence.

The expression used as a score for the unigram matching module was as follows:

$$\text{Bigram Score} = \frac{\text{Number of bigram matching}}{\text{Total number of bigrams in SMS}}.$$

3.4 1-Skip and Inverse Bigram Matching

For each pair of words <s_i, t_j> in the list S′ that is found in the inverse order <t_i, s_j> in F′, we applied various semantic rules, because such pairs can not be just rejected. The complete set of the rules is not given here. For example, if one of the two words in the pair was a verb, then we ignored the difference and considered the bigram sequence as a match.

If there was a negation word in one of the two texts between two given words, but the two words formed a bigram in the other text, then we just removed the negation word and, again, considered the bigram sequence as a match. That is, under some circumstances we considered a sequence of two words with one gap as a bigram.

3.5 Final Ranking

Finally, the overall scoring was calculated as the harmonic mean of the two particular scores:

$$\text{Total Score} = \frac{\text{Unigram Score} \times \text{Bigram Score}}{\text{Unigram Score} + \text{Bigram Score}}.$$

We presented to the user as the output the top five scores for a single SMS query.

4 System Architecture for Task 2: Cross-Lingual

The architecture of our system for Task 2 (cross-lingual task) is presented in Figure 2. Our system for this task has three major modules. The objective of the first module is

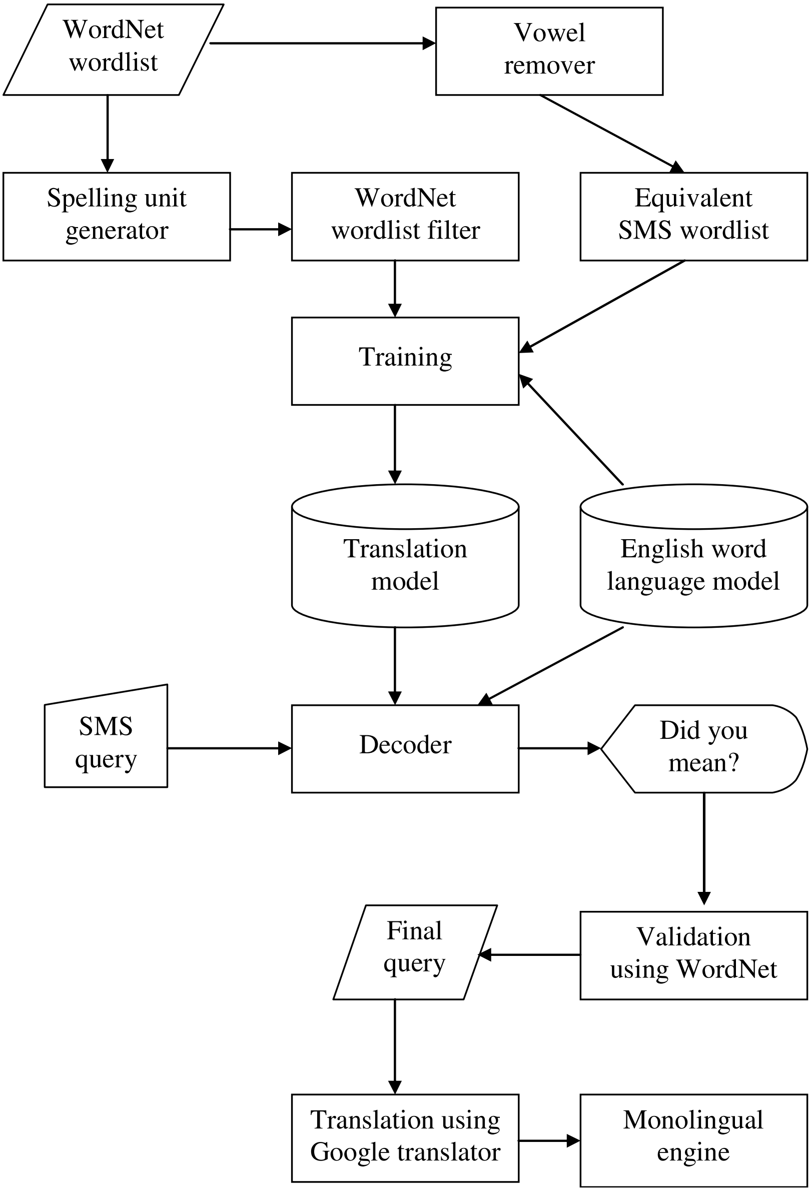


Fig. 2. Cross-lingual FAQ retrieval system Architecture

to compose an SMS query to the monolingual English query translation model. An English correction module corrects the output of the SMS-English query translation system. Finally, the system retrieves the answer from the FAQ dataset by using a cross lingual FAQ retrieval system.

4.1 SMS to English Query Translation Module

In this module, initially, we have all English words known to the system listed in a file. These words have been collected from the English WordNet 3.0 resource. The English wordlist file plays an important role in the development of parallel example-based Hindi SMS—English translation system.

Next, the system trained and decoded the SMS query using the English query translation. This module has the following sub-modules:

SMS-English Parallel Corpus. We have created a database of English words, which were collected from the WordNet database: verbs, nouns, adverbs, and adjectives. We removed the vowels from each word: e.g., a word "Translation" became "Trnsltn". This is important because vowels are often omitted in SMS texts. We stored such words stripped from the vowels along with the original word and the word separated into spelling units, e.g., "tra-ns-la-tion".

Training and Decoding. We have developed an SMS-English word translation model by using a statistical tool: Moses toolkit. The translation model generates English words from SMS words. Given an input of as an SMS text, the system generates the corresponding English word-by-word translation. After decoding SMS query in this way, we form the English query corresponding to this SMS text.

Cleaning. The generated English query is corrected by a spell checker, as discussed in Section 2. For further cleaning the English query, we validated each word against the WordNet resource. If the word is not present in WordNet, then we extracted the nearest candidates of that unknown word. Letters of the unknown word were matched against the nearest candidate word. The matching procedure followed the letter sequence matching.

4.2 English—Hindi Translation System and Final Retrieval

Finally, the generated English query was translated into the Hindi query using freely available online Google translator. With this, the cross-lingual SMS query was obtained. Then, the monolingual FAQ retrieval procedure of Hindi SMS queries described in Section 2 was used.

5 Experimental Results

We tested our SMS-based FAQ retrieval on one run for Monolingual (English), two runs for Monolingual (Hindi), and one run Cross-lingual (Hindi) of the FIRE 2011 competition.

We used the Mean Reciprocal Rank (MRR) as one of the evaluation measures for our system. MRR is non-zero if, for the in-domain queries, a correct match was found in any of their top 5 answers. If the correct answer was returned as the top answer, then the score was the highest for that query; if it was returned as the 5th answer then the score was the lowest.

The evaluation scores obtained by our SMS-based FAQ retrieval are shown in Table 3.

Table 3. Evaluation Score for SMS-based FAQ retrieval

Run	Descriptions	Statistics
Monolingual (English)	# of In-domain Queries	704
	# of Out of Domain Queries	2701
	In Domain Correct	29 / 704 = 0.0412
	Out of Domain Correct	0 / 2701 = 0
	Mean Reciprocal Rank (MRR)	0.0538
Monolingual (Hindi), run 1	# of In-domain Queries	200
	# of Out of Domain Queries	124
	In Domain Correct	0 / 200 = 0
	Out of Domain Correct	119 / 124 = 0.960
	Mean Reciprocal Rank (MRR)	0.0
Monolingual (Hindi), run 2	# of In-domain Queries	200
	# of Out of Domain Queries	124
	In Domain Correct	36 / 200 = 0.180
	Out of Domain Correct	0 / 124 = 0
	Mean Reciprocal Rank (MRR)	0.235
Cross-lingual	# of In-domain Queries	37
	# of Out of Domain Queries	3368
	In Domain correct	2 / 37 = 0.0541
	Out of Domain correct:	40 / 3368 = 0.0119
	Mean Reciprocal Rank (MRR)	0.0541

As it can be observed from Table 3, the system gives promising results, which are above a random baseline, though much lower than what would be required for practical use of the system.

We believe that such low results can be explained by the noisy nature of both SMS queries (which are usually written in a language very different from the norm) and the text of the FAQ answers.

6 Conclusions and Future Work

In this paper we have described our approaches for monolingual and cross-lingual SMS-based FAQ retrieval. For spell checking, we used the freely available Bing spellchecker. For translation, we used the freely available Google translator because Google translator has better performance than the Bing translator.

In our future work we plan to develop our own machine translation system optimized for this kind of tasks. In addition, we plan to explore the use of softer semantic and syntactic [5] similarity measures, such as those that employ argument structure information [6] and information about synonyms [7].

Acknowledgements. The work was partially supported by the Governments of India and Mexico under the CONACYT-DST India (CONACYT 122030) project “Answer Validation through Textual Entailment”, the Government of Mexico under the CONACYT 50206-H project and SIP-IPN 20121823 project through Instituto Politécnico Nacional, and the Seventh Framework Programme of European Union, project 269180 “Web Information Quality Evaluation Initiative (WIQ-EI)”.

References

1. Ledeneva, Y., Sidorov, G.: Recent Advances in Computational Linguistics. Informatica. International Journal of Computing and Informatics 34, 3–18 (2010)
2. FIRE 2011: Third Workshop of the Forum for Information Retrieval Evaluation, IIT Bombay, December 2-4
3. Contractor, D., Mittal, A., Padmanabhan, D.S., Subramaniam, L.V.: SMS-based FAQRetrieval. In: Third Workshop of the Forum for Information Retrieval Evaluation, FIRE 2011, IIT Bombay, December 2-4 (2011)
4. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41
5. Sidorov, G., Herrera-de-la-Cruz, J.A., Galicia-Haro, S.N., Posadas-Durán, J.P., Chanona-Hernandez, L.: Heuristic Algorithm for Extraction of Facts Using Relational Model and Syntactic Data. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part I. LNCS (LNAI), vol. 7094, pp. 328–337. Springer, Heidelberg (2011)
6. Castro-Sánchez, N.A., Sidorov, G.: Analysis of Definitions of Verbs in an Explanatory Dictionary for Automatic Extraction of Actants Based on Detection of Patterns. In: Hopfe, C.J., Rezgui, Y., Métais, E., Preece, A., Li, H. (eds.) NLDB 2010. LNCS, vol. 6177, pp. 233–239. Springer, Heidelberg (2010)
7. Castro-Sánchez, N.A., Sidorov, G.: Automatic Acquisition of Synonyms of Verbs from an Explanatory Dictionary Using Hyponym and Hyperonym Relations. In: Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Ben-Youssef Brants, C., Hancock, E.R. (eds.) MCPR 2011. LNCS, vol. 6718, pp. 322–331. Springer, Heidelberg (2011)

Extrinsic Evaluation on Automatic Summarization Tasks: Testing Affixality Measurements for Statistical Word Stemming

Carlos-Francisco Méndez-Cruz[1,4], Juan-Manuel Torres-Moreno[1,2],
Alfonso Medina-Urrea[3], and Gerardo Sierra[4]

[1] LIA-Université d'Avignon et des Pays de Vaucluse, France
cmendezc@ii.unam.mx, juan-manuel.torres@univ-avignon.fr
[2] École Polytechnique de Montréal, Canada
[3] El Colegio de México A.C., México
amedinau@colmex.mx
[4] GIL-Instituto de Ingeniería UNAM, México
gsierram@ii.unam.mx

Abstract. This paper presents some experiments of evaluation of a statistical stemming algorithm based on morphological segmentation. The method estimates affixality of word fragments. It combines three indexes associated to possible cuts. This unsupervised and language-independent method has been easily adapted to generate an effective morphological stemmer. This stemmer has been coupled with Cortex, an automatic summarization system, in order to generate summaries in English, Spanish and French. Summaries have been evaluated using ROUGE. The results of this extrinsic evaluation show that our stemming algorithm outperforms several classical systems.

Keywords: Automatic summarization, Affixality Measurements, Morphological Segmentation, Statistical Stemming, CORTEX.

1 Introduction

Discovering linguistic units in Natural Languages has been a long-standing human task. Now, automatic approaches are used to conduct this work. Despite rule-based approaches for word processing are widely used, there is a renewed interest in morphological methods. Corpora of morphologically complex languages, such as agglutinative ones, can be processed in order to discover morphemes. Simple strategies are not suitable for this kind of languages. Also, corpora for languages, which have not been computationally studied, appear every day. Then, this paper presents a method for unsupervised learning of morphology.

A morphologist tries to collect morphological units from languages, among other tasks. Roughly, two types of units are considered in this paper: stems and affixes. On the one hand, a stem is the part of a word stripped of affixes, it carries lexical content. On the other hand, affixes carry either grammatical information or information to produce another word. Finally, by its position, affixes can be

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 46–57, 2013.

classified in prefixes, before the stem; and suffixes, following it. In morphologically complex languages, words can be recursively formed by stems and affixes. Thus, this paper deals with discovering stems and suffixes, i.e. morphological segmentation and its evaluation. We propose an extrinsic evaluation by means of a Natural Language Processing (NLP) task. We developed a stemmer to test some procedures to eliminate suffixes from a word. The stemmer is coupled with an automatic text summarizer. Lastly, summaries are evaluated with ROUGE, a semi-automatic evaluation system.

2 State of the Art

2.1 Morphological Segmentation

Morphological segmentation tries to discover morphological units (morphs) from a sequence of symbols of language. This problem is closely dependent on language. Nonetheless, computational approaches have made traditionally simplistic assumptions such as a word has a simple structure as *prefix-stem-suffix* [1].

The first work for unsupervised discovery of morphemes is due to Zellig Harris [2]. His approach consisted in counting distinct symbols preceding/following a possible morphological boundary. Thus, high frequency corresponds to true morphological boundaries. After Harris's work, many approaches have been proposed. A wide review of methods can be seen in [3,4,5]. A large majority of them sees morphology as a lexicon of words. Their aim is to reduce the redundancy in the lexicon. A method that follows this idea was proposed in [6,7]. This employs Minimum Description Length (MDL) analysis as a strategy for obtaining the lowest redundancy in the lexicon. Hence, the best morphology is obtained when the description length of the data is the lowest. To control the quality of segmentations, this method uses combinatorial structures named *signatures*. Also, signatures are involved in calculating the description length. This approach has been implemented in a computational program named Linguistica.[1]

Another method that searches also an optimal morphological model has been presented in [8,1,9,10]. The set of methods developed in these papers has been called *Morfessor*. It has been formulated for agglutinative languages such as Finnish. Regarding [10], it incorporates a morphotactic analysis for each word. Thus, given some initial segmentation, four possible categories are assigned: prefix, suffix, stem, and no-morph. The method uses Maximum a Posteriori (MAP) framework, which is essentially equivalent to MDL. In order to assign those categories, each word is represented by a Hidden Markov Model (HMM).

An effort for obtaining a minimal model of inflectional morphology for Spanish is exposed in [11,12] It uses genetic algorithms for finding the best lexicon. The fitness function is based on MDL principles.

[1] `http://linguistica.uchicago.edu`

2.2 Stemming Works

In general, stemming algorithms aim to remove inflectional and derivational suffixes from words. In many tasks of NLP, such as information retrieval, question-answering, or automatic summarization, stemming is an important part of text preprocessing. Often, a document is represented as a Vector Space Model. Then, in order to improve performance, conflating similar words is preferred.

Thus, language dependent approaches based on handmade rules are widely used, for instance [13]. In fact, the Porter's stemmer [14] is utilized in many NLP systems for European languages. In this approach, a set of removing rules is proposed, where suffixes are substituted by other ones, even the null suffix. A short example from [14] (Step 3) is listed in (1).

(1) a. ICITI → IC (electriciti → electric)
b. ICAL → IC (electrical → electric)
c. FUL → null (hopeful → hope)

However, information requirements in more languages have emerged. Also, agglutinative languages need strategies for stemming different from simply suffix removal. These facts hold research attention in unsupervised approaches, instead of adapting Porter's rules. Furthermore, some works have focused on the disadvantages of rule-based approaches, like [15].

A review of stemming methods can be found in [16]. A language independent approach [17] presents a clustering based unsupervised technique named YASS. A distance function is used in order to measure orthographical similarity, and to assign a low distance value to related words. Other methods which use graphs are [18,19]. The former splits words from a text at all possible positions. Then, segmentations are represented as a directed graph. An algorithm is used to estimate prefix and suffix scores to maximize the probability of a prefix-suffix pair. On the other hand, a method that uses frequency of n-grams of letters as strategy of stemming is presented in [20].

A word regularization process close to stemming is lemmatization. Given a group of words grammatically related, this process selects a representative of them (lemma). However, [21] has stressed some problems in NLP tasks. Also, it presents an unsupervised algorithm based on word clustering by means of a similarity matrix that searches a lemma for a group of words semantically related. Finally, there has been a strong interest in stemming evaluation. For instance, [15] made an evaluation of some stemming algorithms for information retrieval. This work showed that stemming improves performance significantly in short documents. We found similar conclusions in [22], where five stemming algorithms were evaluated using the SMART text retrieval system.

3 Affixality Measurements

A brief description of affixality measurements is presented in this section. In [5,23] these measurements are exposed with more detail for Spanish. Also, its application in unrelated languages could be found in [24] for Czech, and [25] for the Amerindian Languages Chuj and Tarahumara. The idea behind this approach is to quantify the affixality of a word segment. In other words, it tries to estimate the combinatorial *force* of linguistics units. One expects that higher values of that affixality correspond to morphological segmentations. It is clear that, regarding other methods, this one is not guided for searching an optimal morphological model. Actually, three statistical measurements are used for computing the affixality. They are presented below.

3.1 Entropy

Harris's idea revealed that uncertainty is a good indicator of morphological cuts. This idea is closely related to Shannon's concept of information content (entropy) [26]. Therefore, given $a_{i,j}{::}b_{i,j}$ as a word segmentation,[2] and $B_{i,j}$ a set of all segments combined with $a_{i,j}$ the entropy is obtained as follows:

$$H\left(a_{i,j} :: B_{i,j}\right) = -\sum p\left(b_{k,j}\right) \times log_2\left(p\left(b_{k,j}\right)\right) \tag{1}$$

where $k = 1, 2, 3, \ldots |B_{i,j}|$ and each $b_{k,j} \in B_{i,j}$. As it was tested in [23], peaks of entropy from right to left are good indicators of a suffix segmentation.

3.2 Economy Principle

Morphological phenomena work as economical systems. Fewer units are combined at one level in order to create a great number of another units at the next level (Economy Principle). Thus, we can expect that word stems belong to a big set of relatively infrequent units, and affixes to a small set of frequent ones.

In [27] a quantification of this economy was proposed. Here, a reformulation is presented. Given a word segmentation $a_{i,j}{::}b_{i,j}$, let $A_{i,j}$ be the set of segments which alternate with $b_{i,j}$ $(a_{i,j} \in A_{i,j})$, and $B_{i,j}$ a set of segments which alternate with $a_{i,j}$ $(b_{i,j} \in B_{i,j})$. Now, let $A^p_{i,j}$ be the set of segments which are likely prefixes, and $B^s_{i,j}$ the set of segments which are likely suffixes. Thus, the economy of a segmentation is formulated in two ways, depending on type of morph hypothesized:

$$K^p_{i,j} = 1 - \frac{|A_{i,j}| - |A^p_{i,j}|}{|B^s_{i,j}|}; \qquad K^s_{i,j} = 1 - \frac{|B_{i,j}| - |B^s_{i,j}|}{|A^p_{i,j}|} \tag{2}$$

3.3 Numbers of Squares

A square is found in language when four expressions, let say A, B, C, D, are combined to form AC, BC, AD, and BD. This concept was proposed by Joseph Greenberg in [28]. An example in Spanish would be *abarc::aba*, *camin::aba*, *abarc::aron*, *camin::aron*. Hence, let $c_{i,j}$ be the number of squares found in segment j of the word i.

[2] We use :: as a segmentation mark.

3.4 Measurements Combined

Then, affixality is estimated by a combination of the three measurements previously explained. Consequently, an average of normalized values is calculated as a combination strategy:

$$AF^n\left(s_x\right) = \frac{c_x/\max\ c_i + k_x/\max\ k_i + h_x/\max\ h_i}{3} \tag{3}$$

An important fact is that no explicit distinction between inflection and derivation is involved in the procedure of affixality calculation. In addition, affixality could be obtained either from left to right or conversely. Left to right affixality permits us to discover prefixes, while right to left affixality fits in better with suffixes. For calculating these measurements only raw text is necessary as a training corpus. In our experiments we will evaluate three differents sizes (see section 5.1). An example of the affixality index for a Spanish word is shown in Table 1. The affixality index has been used in previous work for gathering affix catalogs [5,23]. The idea is to split a word in two segments, taking the highest value as the cut. For instance, in Table 1 the cut occurs between the stem UTILIZADO∼ and the suffix ∼S (UTILIZADO::S). Our first interest is proposing a segmentation strategy which leads us to discover all morphs (in this paper suffixes).

Table 1. Affixality for Spanish word: `UTILIZADOS` (masculine and plural form of past participle of verb TO UTILIZE)

U	T	I	L	I	Z	A	D	O	S
« 0.079	0.0698	0.1291	0.1566	0.3066	0.7137	0.1556	0.8097	0.8231	

« Right to left affixality

4 Methodology

In this section, we describe a simple approach, based on the affixality index to morphological segmentation. Then, we propose an extrinsic evaluation strategy by state-of-art automatic summarizer.

4.1 Morphological Segmentation by Affixality Measurements

Once the affixality index has been obtained, we can choose different ways to discover morphological segments. Regarding that, [23] propounded four strategies: (1) *max(affixality)*, (2) *affixality > a threshold*, (3) *affixality > 0*, and (4) *max(affixality) recursively*. Here, we use a simple peak-valley strategy for segmentation. Given a set of affixality indexes inside a word af_i^k, let $af_{i-1}^k < af_i^k > af_{i+1}^k$ be a peak of affixality from left to right, where k is the length of the word plus one: the ending of the word.

Furthermore, we used two heuristic strategies. First, we start searching for a peak of affixality at $i = 3$. Second, the ending of a word has zero affixality, i.e. $af_k = 0$. The first assumption let us avoid stem over-segmentation, whereas the second assumption allows one-letter suffixes at the end of a word. For instance, from Table 1 we obtained the segmentation UTILIZ::ADO::S. There are some disadvantages to this approach. For example, slight peaks could be taking into account as possible morphological cuts. Once the new morphological segmentation strategy is proposed, an evaluation will be required. There are two criteria for evaluation: intrinsic and extrinsic [29]. Intrinsic evaluation for morphological segmentation requires comparison to a morphological gold standard. Despite it is possible to find some handmade segmentation corpus, we suggest for this paper an extrinsic evaluation.

A natural purpose for morphological segmentation is stemming. In [30] highest affixality values were suggested for word truncation. The proposed evaluation showed that this approach outperforms the Porter's stemmer. In that paper, the affixality was calculated only with economy and entropy measurements. In this paper, we additionally use squares. Alternatively, our goal consists in testing two strategies that generate two possible cuts. First, truncating at the rightmost peak of the word (UTILIZ::ADO::S > UTILIZADO∼) would result in inflectional stemming. Second, truncating at leftmost peak (UTILIZ::ADO::S > UTILIZ∼) would result in derivational-inflectional stemming.

4.2 Automatic Summarization Like Extrinsic Evaluation Task

An automatic summarization task was selected for this extrinsic evaluation. Particularly, the Cortex system was chosen for this purpose. Its modular architecture allowed us to adapt easily the new stemmer. Preprocessing texts is the first step in Cortex. Texts are filtered, and words are lemmatized by means of a dictionary. The number of words in the dictionary is for English 97K, for Spanish 624K and for French 194K. Instead of this lemmatization stage, the Porter's stemmer could be used.[3] Even both processes could be sequentially applied. In this step, our stemmer has been coupled with Cortex.

4.3 Cortex **Summarizer**

An automatic text summarizer extracts the most relevant sentences from a text [31]. The Cortex system generates summaries from texts represented in Vector Space Model. A decision algorithm combines several metrics in order to score sentences. After filtering, a frequency matrix γ is constructed: each element γ_i^μ of this matrix represents the number of occurrences of the word i in the sentence μ; $1 \leq i \leq M$ words, $1 \leq \mu \leq P$ sentences. Matrix ξ represents the presence/absence of terms in a sentence μ.

[3] Specifically, a Perl implementations from Lingua-Stem-Snowball, `http://snowball.tartarus.org/`

$$\gamma = \begin{bmatrix} \gamma_1^1 & \gamma_2^1 & \cdots & \gamma_i^1 & \cdots & \gamma_M^1 \\ \gamma_1^2 & \gamma_2^2 & \cdots & \gamma_i^2 & \cdots & \gamma_M^2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_1^P & \gamma_2^P & \cdots & \gamma_i^P & \cdots & \gamma_M^P \end{bmatrix}, \quad \gamma_i^\mu \in \{0, 1, 2, \dots\} \tag{4}$$

4.4 The Metrics

Important mathematical and statistical information can be extracted from the matrices γ and ξ. CORTEX uses Γ metrics calculated on frequencies, entropy, Hamming and hybrid values. See [32] for more information of CORTEX's metrics.

1. **Frequency measures**
   (a) Term Frequency: $F^\mu = \sum_{i=1}^{M} \gamma_i^\mu$
   (b) Interactivity of segments: $I^\mu = \sum_{\substack{i=1 \\ \xi_i^\mu \neq 0}}^{M} \sum_{\substack{j=1 \\ j \neq \mu}}^{P} \xi_i^j$
   (c) Sum of probability frequencies: $\Delta^\mu = \sum_{i=1}^{M} p_i \gamma_i^\mu ; p_i =$ word's i probability
2. **Entropy.** $E^\mu = -\sum_{\substack{i=1 \\ \xi_i^\mu \neq 0}}^{M} p_i \log_2 p_i$
3. **Measures of Hamming.** These metrics use a Hamming matrix H, a square matrix $M \times M$:

$$H_n^m = \sum_{j=1}^{P} \left\{ \begin{array}{ll} 1 & \text{if } \xi_m^j \neq \xi_n^j \\ 0 & \text{elsewhere} \end{array} \right\} \quad \text{for} \quad \begin{array}{l} m \in [2, M] \\ n \in [1, m] \end{array} \tag{5}$$

   (a) Hamming distances: $\Psi^\mu = \sum_{\substack{m=2 \\ \xi_m^\mu \neq 0}}^{M} \sum_{\substack{n=1 \\ \xi_n^\mu \neq 0}}^{m} H_n^m$
   (b) Hamming weight of segments: $\phi^\mu = \sum_{i=1}^{M} \xi_i^\mu$
   (c) Sum of Hamming weight of words per segment: $\Theta^\mu = \sum_{\substack{i=1 \\ \xi_i^\mu \neq 0}}^{M} \psi_i$; every word. $\psi_i = \sum_{\mu=1}^{P} \xi_i^\mu$
   (d) Hamming heavy weight: $\Pi^\mu = \phi^\mu \Theta^\mu$
   (e) Sum of Hamming weights of words by frequency: $\Omega^\mu = \sum_{i=1}^{M} \psi_i \gamma_i^\mu$
4. **Titles.** $\theta^\mu = \cos\left(\frac{\sum_{i=1}^{M} \gamma_i^\mu \text{Title}}{\|\gamma^\mu\| \|\text{Title}\|} \right)$

4.5 Decision Algorithm (DA)

The Decision Algorithm combines all normalized sentence scores. Two averages are calculated: $\lambda_\mu > 0.5$, and $\lambda_\mu < 0.5$ ($\lambda_\mu = 0.5$ is ignored):

$$\sum^{\mu} \alpha = \sum_{\substack{\nu=1 \\ \|\lambda_\mu^\nu\| > 0.5}}^{\Gamma} \left(\| \lambda_\mu^\nu \| - 0.5 \right) ; \sum^{\mu} \beta = \sum_{\substack{\nu=1 \\ \|\lambda_\mu^\nu\| < 0.5}}^{\Gamma} \left(0.5 - \| \lambda_\mu^\nu \| \right) \tag{6}$$

ν is the index of the metrics, $\sum_{\nu}^{\Gamma}$ is the sum of the absolute differences between $\|\lambda\|$ and 0.5, $\sum^{\mu} \alpha$ are the "positive" normalized metrics, $\sum^{\mu} \beta$ the "negative" normalized metrics and Γ the number of metrics used. The score of each sentence is calculated as follows:

$$\text{If } \left(\sum^{\mu} \alpha > \sum^{\mu} \beta \right)$$

$$\text{then } \Lambda^{\mu} = 0.5 + \frac{\sum^{\mu} \alpha}{\Gamma} \text{ else } \Lambda^{\mu} = 0.5 - \frac{\sum^{\mu} \beta}{\Gamma}$$

All sentences are sorted using $\Lambda^{\mu}; \mu = 1, \cdots, P$. The compression rate τ determines the final size of the summary.

5 Experiments and Results

5.1 Design of Experiments

In order to evaluate our stemmer, we performed some experiments involving (i) three different stemming strategies, (ii) corpora in different languages: English, Spanish and French, and (iii) different sizes of training corpora: 100K, 200K, and 500K word tokens. Our stemmer is compared to Cortex's lemmatizer (lemm), the Porter's stemmer used by Cortex (stem), and both methods sequentially applied (lems). In addition, we included no stemming at all (raw) and fixed truncation at 6 characters (fixed). Respecting to our stemmer, we tested three possible cuts of a word based on morphological segmentation, all of them with the three sizes of training corpora: highest affixality value (vM100, vM200, vM500), first peak of affixality at right (R100, R200, R500), and first peak of affixality at left (L100, L200, L500).

Summaries were evaluated using ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) [33]. This semi-automatic evaluation system calculates a score of similarity between a candidate summary and several human summaries. We evaluated them using bigrams (ROUGE-2) and skip bigrams (ROUGE-SU4).

5.2 Corpora

Three sets of documents were selected for the evaluation in English, Spanish and French. For English, 50 clusters from DUC 2004[4] were used, specifically the *Task 2 - Short multi-document summaries focused by TDT events.* The clusters contained on average 10 documents from the AP newswire and the New York Times. Four human summaries per cluster were available for summaries evaluation. For Spanish, 8 biomedical articles were collected from the specialized journal *Medicina Clínica.*[5] In this case, we evaluated automatic summaries against author's

[4] http://duc.nist.gov/duc2004

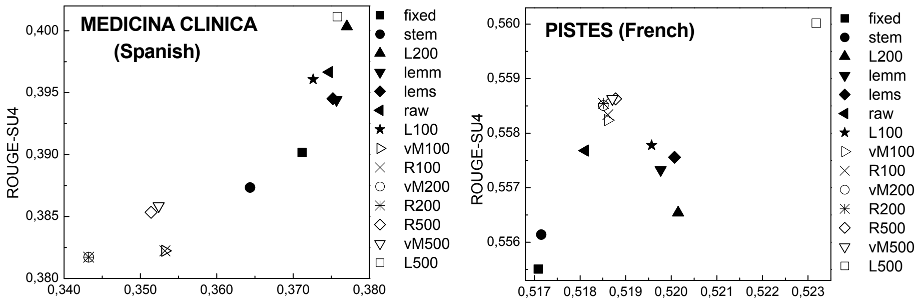
[5] http://www.elsevier.es/revistas/ctl_servlet?_f=7032&revistaid=2

abstracts. Regarding French evaluation, we utilized *Canadien French Sociological Articles* corpus [34] from the specialized e-journal *Perspectives interdisciplinaires sur le travail et la santé* (PISTES).[6] 50 sociological articles constituted this corpus. One human abstract per text was used for evaluation.

Regarding training corpora, we used different texts than those in the evaluation task. We constituted three sets of documents of 100K, 200K and 500K word tokens per each language. For English, we selected 24 documents from INEX 2012 *Tweet Contextualization Track.*[7] Each document was formed by passages from Wikipedia that contextualized a tweet query. For Spanish, we used the *Corpus del Español Mexicano Contemporáneo, CEMC* [35], a well-balanced text collection from different sources. Finally, for the French corpus we gathered texts from several sources. First, two corpora from *DÉfi Fouille de Textes* (DEFT'07) were included. Second, we used *aVoiraLire* (3 460 critiques of books, films, comics and shows) and *relectures* (1 484 reviews of scientific articles). Also, the book *Pensée, essais, maximes et correspondance de J. Joubert* was included. At last, 2 472 phrases of texts of Jules Verne were considered.

5.3 Results

The ROUGE metrics obtained on the three corpora are plotted in Figures 1 and 2. For Spanish (Figure 1(a)) and French (Figure 1(b)), we can observe that the method L500 obtained the best score.



(a) Spanish corpus

(b) French corpus

Fig. 1. Content evaluation for Spanish and French corpus

On the other hand, first peak of affixality at right and the highest value of affixality methods obtained lower scores. It means that CORTEX performs well for Spanish and French with a derivational-inflectional stemmer. By comparison, this method (L500) was the worst on English corpus (see Figure 2). In fact, the

[6] `http://www.pistes.uqam.ca/index.html`

[7] `https://inex.mmci.uni-saarland.de/tracks/qa/`

truncating methods at the end of the word appeared more successful, for instance vM100, R100. This suggests a correlation between the segmentation approach and the morphological complexity.

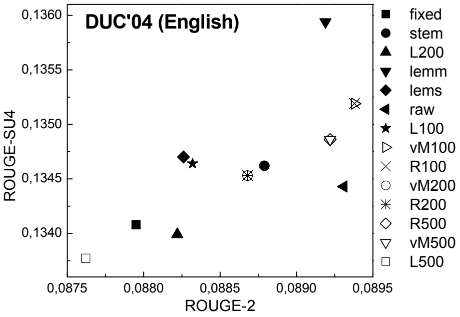


Fig. 2. Content evaluation for English corpus

Surprisingly, the raw method performed as good as Cortex's strategies (lemm, lems), and overcame stem and some affixality approaches for Spanish. Besides, in this language also the fixed method was better than the Porter's stemmer. Similar situation was found in French (see Figure 1(b)), where the raw method defeated the stem method. Consequently, Cortex works better while word normalization increases. This could explain the first position of lemm (a dictionary method) for English.

Regarding corpus sizes, position of L500 in Spanish and French evaluation proposes that an improvement is obtained increasing the size of the training corpus. Nevertheless, for English it is not true. We can explain this because of the relatively simple inflectional morphology of this language. Lastly, we can observe for all languages coincidences between the highest affixality method and first peak method of affixality at right. This situation is due to the fact that the most affixality segments are inflection suffixes, in fact, the leftmost suffixes.

6 Conclusions and Future Work

We have presented a study of statistical stemming through morphological segmentation. This unsupervised and language-independent approach uses affixality measurements for unsupervised learning of morphology. Our results confirm a correlation between the stemming strategy and the morphological complexity of a language. It means that the rule-based stemming loses effectiveness as the morphological complexity increases. In this manner, our statistical stemmer outperforms, for Spanish and French, Porter's and based-dictionary stemming strategies. In consequence, morphological stemmers should be taken into account.

Regarding morphological segmentation, there is always room for some improvement. Different strategies for segmenting should be tested. Even though, results of extrinsic evaluation by means of automatic summarization seem to be promising, an intrinsic evaluation against a gold standard should be done. In the future we will evaluate our method by using FRESA [34,36], it will allow us to test in different corpora avoiding having human summaries.

Acknowledgments. We thank El Colegio de México A.C. for allowing us to use the CEMC corpus. This work was partially supported by a CONACyT grant.

References

1. Creutz, M., Lagus, K.: Unsupervised Discovery of Morphemes. In: Proc. of the Workshop on Morphological and Phonological Learning of ACL 2002, Philadelphia, SIGPHON-ACL, pp. 21–30 (2002)
2. Harris, Z.S.: From Phoneme to Morpheme. Language 31, 190–222 (1955)
3. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. ACM Trans. Speech Lang. Process 4 (2007)
4. Goldsmith, J.A.: Segmentation and Morphology. In: The Handbook of Computational Linguistics and Natural Language Processing, pp. 364–393. Wiley-Blackwell, Oxford (2010)
5. Medina-Urrea, A.: Investigación cuantitativa de afijos y clíticos del español de México. Glutinometría en el Corpus del Español Mexicano Contemporáneo. PhD thesis, El Colegio de México, México (2003)
6. Goldsmith, J.: Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics 27, 153–198 (2001)
7. Goldsmith, J.: An Algorithm for the Unsupervised Learning of Morphology. Natural Language Engineering 12, 353–371 (2006)
8. Creutz, M.: Unsupervised segmentation of words using prior distributions of morph length and frequency. In: Hinrichs, E., Roth, D. (eds.) 41st Annual Meeting of the ACL, Sapporo, Japan, pp. 280–287 (2003)
9. Creutz, M., Lagus, K.: Induction of a Simple Morphology for Highly-Inflecting Languages. In: Proc. of 7th Meeting of the ACL Special Interest Group in Computational Phonology SIGPHON-ACL, pp. 43–51 (2004)
10. Creutz, M., Lagus, K.: Inducing the Morphological Lexicon of a Natural Language from Unannotated Text. In: Int. and Interdisciplinary Conf. on Adaptive Knowledge Representation and Reasoning (AKRR 2005), pp. 106–113 (2005)
11. Gelbukh, A., Alexandrov, M., Han, S.-Y.: Detecting Inflection Patterns in Natural Language by Minimization of Morphological Model. In: Sanfeliu, A., Martínez Trinidad, J.F., Carrasco Ochoa, J.A. (eds.) CIARP 2004. LNCS, vol. 3287, pp. 432–438. Springer, Heidelberg (2004)
12. Reyes, D.: Sistema de segmentación automática de palabras para el español. Master's thesis, CIC-IPN (2008)
13. Lovins, J.B.: Development of a Stemming Algorithm. Mechanical Translation and Computational Linguistics 11, 23–31 (1968)
14. Porter, M.F.: An algorithm for Suffix Stripping. Program 14, 130–137 (1980)
15. Krovetz, R.: Viewing Morphology as an Inference Process. In: Proccedings of the 16th ACM/SICIR Conference, pp. 191–202 (1993)

16. Lennon, M., Pierce, D., Tarry, B., Willet, P.: An evaluation of some conflation algorithms for information retrieval. J. of Information Science 3, 177–183 (1981)
17. Majumder, P., Mitra, M., Pal, D.: Bulgarian, Hungarian and Czech Stemming Using YASS. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 49–56. Springer, Heidelberg (2008)
18. Bacchin, M., Ferro, N., Melucci, M.: A probabilistic model for stemmer generation. Mechanical Translation and Computational Linguistics 41, 121–137 (2005)
19. Paik, J.H., Mitra, M., Parui, S.K., Jarvelin, K.: GRAS: An effective and efficient stemming algorithm for information retrieval. ACM Trans. Inf. Syst. 29 (2011)
20. McNamee, P., Mayfield, J.: Character n-gram tokenization for European language text retrieval. Information Retrieval 7, 73–97 (2004)
21. Torres-Moreno, J.M.: Reagrupamiento en familias y lexematización automática independientes del idioma. Inteligencia Artificial 47, 38–53 (2010)
22. Hull, D.A.: Stemming algorithms - A case study for detailed evaluation. Journal of the American Society for Information Science 47, 70–84 (1996)
23. Medina-Urrea, A.: Automatic Discovery of Affixes by means of Corpus: A Catalog of Spanish Affixes. Journal of Quantitative Linguistics 7, 97–114 (2000)
24. Medina-Urrea, A., Hlaváčová, J.: Automatic Recognition of Czech Derivational Prefixes. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 189–197. Springer, Heidelberg (2005)
25. Medina-Urrea, A.: Affix Discovery based on Entropy and Economy Measurements. Texas Linguistics Society 10, 99–112 (2008)
26. Shannon, C., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1949)
27. de Kock, J., Bossaert, W.: Introducción a la lingüística automática en las lenguas románicas. Gredos, Madrid (1974)
28. Greenberg, J.H.: Essays in Linguistics. The Univ. of Chicago Press, Chicago (1957)
29. Spärck-Jones, K., Galliers, J.: Evaluating Natural Language Processing Systems: An Analysis and Review. Springer, New York (1996)
30. Medina-Urrea, A.: Towards the Automatic Lemmatization of 16th Century Mexican Spanish: A Stemming Scheme for the CHEM. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 101–104. Springer, Heidelberg (2006)
31. Torres-Moreno, J.M.: Résume automatique de documents, Lavoisier, Paris (2011)
32. Torres-Moreno, J.M., St-Onge, P.L., Gagnon, M., El-Bèze, M., Bellot, P.: Automatic Summarization System coupled with a Question-Answering System (QAAS). CoRR abs/0905.2990 (2009)
33. Lin, C.Y.: Rouge: A Package for Automatic Evaluation of Summaries. In: Workshop on Text Summarization Branches Out (2004)
34. Saggion, H., Torres-Moreno, J.M., da Cunha, I., SanJuan, E.: Multilingual summarization evaluation without human models. In: 23rd Int. Conf. on Computational Linguistics, COLING 2010, pp. 1059–1067. ACL, Beijing (2010)
35. Lara, L., Ham Chande, R., García Hidalgo, M.: Investigaciones lingüísticas en lexicografía. El Colegio de México, A.C., México (1979)
36. Torres-Moreno, J.M., Saggion, H., da Cunha, I., SanJuan, E., Velázquez-Morales, P.: Summary Evaluation with and without References. Polibits 42, 13–19 (2010)

Extracting Domain-Specific Opinion Words for Sentiment Analysis

Ivan Shamshurin

National Research University – Higher School of Economics,
20 Myasnitskaya Ulitsa, Moscow, 101000, Russia
ivanshamshurin@gmail.com

Abstract. In this paper, we consider opinion word extraction, one of the key problems in sentiment analysis. Sentiment analysis (or opinion mining) is an important research area within computational linguistics. Opinion words, which form an opinion lexicon, describe the attitude of the author towards certain opinion targets, i.e., entities and their attributes on which opinions have been expressed. Hence, the availability of a representative opinion lexicon can facilitate the extraction of opinions from texts. For this reason, opinion word mining is one of the key issues in sentiment analysis. We designed and implemented several methods for extracting opinion words. We evaluated these approaches by testing how well the resulting opinion lexicons help improve the accuracy of methods for determining the polarity of the reviews if the extracted opinion words are used as features. We used several machine learning methods: SVM, Logistic Regression, Naïve Bayes, and KNN. By using the extracted opinion words as features we were able to improve over the baselines in some cases. Our experiments showed that, although opinion words are useful for polarity detection, they are not sufficient on their own and should be used only in combination with other features.

Keywords: Sentiment Analysis, Natural Language Processing, Machine Learning.

1 Introduction

In this paper, we consider opinion word extraction, one of the key problems in sentiment analysis. Sentiment analysis (or opinion mining) is an important research area within computational linguistics. It is mainly concerned with methods for determining the attitude of the author towards the subject of her text (so-called "polarity") by classifying documents as positive, negative, or neutral.

The increasing popularity of sentiment analysis is due to widespread opinion-rich user-generated resources such as online social networks, personal blogs, wikis, and review websites. [8] provides a general survey of topics in sentiment analysis, including the problem of determining the polarity of texts.

People often express their opinions of products, events, etc. by using subjective opinion words, such as "beautiful", "boring", "interesting", "banal", etc. Opinion

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 58–68, 2013.

words, which form an opinion lexicon, describe the attitude of the author towards certain opinion targets, i.e., entities and their attributes on which opinions have been expressed. Hence, the availability of a representative opinion lexicon can facilitate the extraction of opinions from texts. For this reason, opinion word mining is one of the key issues in sentiment analysis [4, 7, 10–12].

In our paper we applied several methods of machine learning and used opinion words as features in order to determine text polarity (see section 5). The intuition suggests that using opinion words as features makes it possible to improve accuracy of documents classification.

The main issue in creating a list of opinion words is its dependency on the subject domain, e.g., a word can be used to express an opinion in one domain (*"original **Japanese** quality"*) and be neutral in another (*"**Japanese** literature"*). Also there is another case, when a word is an indicator of an opinion in both domains, but in the first one it is positive ("***old*** wine") and in the second one it is negative (*"an **old** car"*).

Hence when forming an opinion word list a better approach is to build domain-specific lists rather than one general-purpose list in the subject domain under consideration. Another reason for this is the fact that some opinion words are created by users and are not contained in dictionaries.

It is worth mentioning that opinion words can be divided into two types: "pure" opinion words (*"beautiful", "boring"*) and "conditional" opinion words, which indicate objective information (size, age, etc.: *"Japanese", "old"*), but which are used as estimative words. Furthermore, the polarity of "pure" opinion words does not depend on the context, whereas in the case of "conditional" opinion words the polarity may change to the opposite one (see example above: *"old wine", "an old car"*).

When extracting "conditionally" evaluative opinion words one should take into account the following fact. Among these adjectives there are both words with constant polarity in the subject domain under consideration, and words with variable polarity. Let us consider the case, when in a certain domain the term $word_1$ is generally used in a positive context, so we label this word as opinion word (i.e. positive). At the same time the term $word_2$ is used both in positive contexts and in negative contexts. The term $word_2$ is opinion word (contains an affective evaluation), but there is approximately equal number of supporters and opponents of $word_2$. In this case it is harder to extract $word_2$ than $word_1$, because the simple statistical approach does not work. In this paper we do not extract this type of opinion words.

2 Related Work

Existing approaches to opinion word extraction can be divided into two categories: corpora-based [3, 5, 15] and dictionary-based [4, 6, 14] methods. We follow the corpora-based paradigm, which makes it possible to extract domain-specific opinion words.

In [3] the method for determining polarities of opinion adjectives using corpus was proposed for the first time. Different conjunction patterns were studied with conjunctions such as *"and", "or"*. The idea is that we can define the polarity of the words in conjoined pairs if we know the polarity of the second word in the pair. As this method relies on conjunctions, the algorithm does not allow the extraction of isolated, not conjoined opinion words.

In [13] a double-propagation method is proposed which outperforms some of state-of-the-art methods. The main idea of the double propagation approach is the following: we start by fixing a set of known opinion words, which will be used as a seed in the subsequent process. Then on each iteration we extract new opinion words (and opinion targets) using words from this seed, as well as words extracted in previous iterations, through some predefined syntactic relations. One of the advantages of this method is that it does not require any additional text corpora and dictionaries. The polarity assignment of the newly extracted words is also implemented.

[16] is the first work in which the task of extracting opinion words-nouns is considered. There it is proposed to determine an opinion word-noun as a noun which can be found more often either in a positive or in a negative context. By context we mean neighborhood of the term (neighbour words), which may contain positive or negative opinion words-adjectives (we assume that evaluative adjectives are known).

In [1] a machine learning based approach for automatic extraction of opinion words (both adjectives and non-adjectives) is proposed. The method requires additional two corpora: a corpus of the neutral descriptions of opinion targets and neutral contrast corpus (news). There are 17 attributes for machine learning, which depend on the corpora. One of them, the deviation index, is used in this paper.

In this paper, we considered several methods for automatic extraction of domain-specific opinion words. For one of these methods, double propagation [13], we proposed a new technique for building a seed to be used as a starting point for opinion word extraction. Syntactic rules of this method were adapted for Russian.

In order to evaluate the methods of opinion word extraction we used several machine learning methods for determining text polarity. For this, we developed several text representations using various features and compared them empirically.

3 Methods of Opinion Word Extraction

We designed and implemented several methods for extracting opinion words.

3.1 Double Propagation Approach

The first approach is based on the double propagation technique proposed in [13], where it is claimed to outperform some of the state-of-the-art methods.

We devised several syntactic rules and tested the approach on reviews in Russian using the Semantic analyzer from AOT project[1] as a dependency parser. Rules for opinion words (OW) and opinion targets (OT) extraction are listed below. Most of them are based on the ideas of the rules from [13].

1. $OW \rightarrow OT$
   (a) $OW - OT$, table 1

Table 1. WT_1 rule

RuleID	Semantic Pattern
WT_1	*OW – PROPERTY – OT*
Example (transliteration): *"V etom filme igraet izvestnaya aktrisa."* Example (translation): *"A famous actress played in this movie."* Output: *famous → actress.* Having *"famous"* we obtain *"actress"*.	

   (b) $OW - H - OT$, i.e. OW depends on OT through H, table 2

Table 2. WT_2 rule

RuleID	Semantic Pattern
WT_2	*OT – F-ACT/S-ACT – IS/ARE – S-ACT/F-ACT – OW*
Example (transliteration): *"Film klasny." "Film – klasny."* These two sentences mean *"The movie is great."* Output: *great → movie.*	

   Here *F-ACT*, *S-ACT* are semantic roles (first and second actants respectively[2]). Here, as we often can see in Russian, the predicate is omitted, but the semantic analyzer understands it and adds *"est"*, which means *"is"/"are"*. So we can identify this kind of relations.
2. $OT \rightarrow OW$
   (a) $OT - OW$, table 3.
   (b) $OT - H - OW$, i.e. OT depends on OW through H, table 4.
3. $OW \rightarrow OW$
   (a) $OW - OW$, table 5.
4. $OT \rightarrow OT$
   (a) $OT - OT$, table 6.
   BELNG is a semantic variable, which means one thing is the part of another one. In this example this relation is identified not as simple as

[1] `http://aot.ru`

[2] `http://aot.ru/docs/SemRels.htm`

Table 3. TW_1 rule

RuleID	Semantic Pattern
TW_1	the same as in WT_1
Example: the same as in WT_1 rule Output: *actress* → *famous.*	

Table 4. TW_2, TW_3 and TW_4 rules

RuleID	Semantic Pattern
TW_2	the same as WT_2
Example: the same as in WT_2. Output: *movie* → *great.*	
TW_3	$OT - CONJ(OW_1, OW_2)$
Example (transliteration): *"Istoriya neobychnaya i intriguyshaya."* Example (translation): *"The story is unusual and intriguing."* Output: *story* → *unusual, intriguing.*	
TW_4	OT – $S\text{-}ACT/F\text{-}ACT$ – IS – $F\text{-}ACT/S\text{-}ACT$ – $AND(OW_1, OW_2)$
Example (transliteration): *"Film horoshy i interesny."* Example (translation): *"The movie is good and interesting."* Output: *movie* → *good, interesting.*	

Table 5. WW_1 and WW_2 rules

RuleID	Semantic Pattern
WW_1	$CONJ(OW_1, OW_2, ...OW_n)$
Example (transliteration): *"Smeshnaya i neobychnaya komediya."* Example (translation): *"A funny and unusual comedy."* Output: *funny* → *unusual.*	
WW_2	$COMMA(OW_1, OW_2, ...OW_n)$
Example (transliteration): *"Krasivaya, zhivaya, rozhdestvenskaya komediya."* Example (translation): *"Nice, lively, Christmas story."* Output: *nice* → *lively, Christmas.*	

.

in English, because in English we use "of the" (which can help identify the relation), but, in Russian, possessive case is used for this purpose, which does not require prepositions.

(b) $OT - H - OT$, table 7.
In the TT_4 examples in Russian sentences the predicate is omitted again, but the semantic analyzer understands it and adds "IS".

Table 6. TT_1 and TT_2 rules

RuleID	Semantic Pattern
TT_1	$CONJ(OT_1, OT_2, ...OT_n)$
Example (transliteration): *"Horoshie rezhiser i aktyori."* Example (translation): *"A good director and actors."* Output: *artist* → *director.*	
TT_2	$OT - BELNG - OT$
Example (transliteration): *"Interesny suzhet filma..."* Example (translation): *"An interesting plot of the movie..."* Output: *movie* → *plot.*	

Table 7. TT_3 and TT_4 rules

RuleID	Semantic Pattern
TT_3	$OT - HAS - OT$
Example (transliteration): *"Rezhiser imeet horoshuyu filmografiu."* Example (translation): *"The director has a good filmography."* Output: *director* → *filmography.*	
TT_4	$OT-F\text{-}ACT/S\text{-}ACT-IS-S\text{-}ACT/F\text{-}ACT-OT$
Example (transliteration): *"Titanic eto luchyiy film." "Titanic – luchyiy film."* The translation of these sentences is *"The "Titanic" is the best movie."* Output: *movie* → *Titanic.*	

3.2 A Method Based on Conditional Probability

This approach is based on conditional probability: if the ratio between the conditional probability of the word occurrence in a positive (negative) review and the conditional probability of the word occurrence in a negative (positive) review is higher than a certain threshold, we label it as an opinion word. In our experiments, we observed that the optimal threshold was 1.25.

3.3 A Method Based on a Pointwise Mutual Information

We also tried two other scoring methods to select opinion words: one is based on the Pointwise Mutual Information [2].

Semantic orientation is based on the concept Pointwise Mutual Information (PMI) with the words "excellent" and "poor".

$$PMI(word_1, word_2) = log_2 \frac{p(word_1 + word_2)}{p(word_1)p(word_2)} \tag{1}$$

$$SO(phrase) = PMI(phrase, \text{``}excellent'') - PMI(phrase, \text{``}poor'') \tag{2}$$

PMI-IR:

$$SO(phrase) = log_2 \frac{hits(phrase\ NEAR\ “excellent'')hits(“poor'')}{hits(phrase\ NEAR\ “poor'')hits(“excellent'')} \tag{3}$$

3.4 An Approach Based on a Deviation Index.

Deviation score based approach reflects the deviation of the average scores of the reviews where the word occurs from the average score of the entire dataset.

$$dev(l) = |\frac{\sum_{i=1}^{n} m_i k_i}{k} - \frac{\sum_{i=1}^{n} m_i}{n}|, \sum_{i=1}^{n} k_i = k \tag{4}$$

Here l is the considering term, n is the total number of reviews, m_i is the mark of the i-th review, k_i is the number of occurrences of the term in i-th review (0 if not occured).

4 Experimental Evaluation

4.1 Data Description

We used the four approaches to extract opinion lexicons from a real-life dataset that was obtained from `http://imhonet.ru/`. It is a customer review collection dated from January 2009 to December 2011 (87563 reviews). We evaluated these approaches by testing how well the resulting opinion lexicons help improve the accuracy of methods for determining the polarity of the reviews if the extracted opinion words are used as features. For this purpose, we collected reviews dated from January 2012 to March 2012 (500 positive and 500 negative reviews) each consisting of at least 15 words.

4.2 Determining Text Polarity

We used several machine learning methods: SVM, Logistic Regression, Naïve Bayes, and KNN, implemented as part of an open-source data mining tool RapidMiner[3].

We represent each document *doc* by a feature-count vector $(n_1(doc),...n_m(doc))$. Our experiments show that machine learning with the presence-based features provide us better results than the machine learning with frequency based features. Presence based features mean setting $n_i(doc)$ to 1 if feature f appears in *doc*, 0 - otherwise.

Baselines. Random-choice baseline provides us with the accuracy 0.5. Accuracies for other baselines are presented below (the average ten-fold cross-validation results):

Table 8. Baselines

	features	# features	Freq./Pres.	KNN	NB	SVM	LR
1	random-choice	0.5					
2	unigrams (2, 2)	741	Freq.	93.4	74.2	85.2	94.3
3	unigrams (2, 2)	741	Pres.	**95.2**	72.7	86.5	95.0
4	*OW* 15 pos, 15 neg	30	Pres.	66.2	59.8	72.5	72.7

Table 9. Manually created list of opinion words (translation presented)

Positive	Negative
beautiful wonderful delightful nice excellent perfect positive magnificent ideal cool high-quality worthful good fine superb	*dim scandalous vile dreadful ugly odious primitive disgusting cheerless idiotic bad vapid slight miserable poor*

In this table features of baselines 2 and 3 are unigrams which occured at least 2 times in positive and 2 times in negative reviews; features of the baseline 4 are manually created 15 positive and 15 negative opinion words (see Table 9).

The baseline feature representations—unigrams and human-selected opinion words—allow us to get 95.2% and 72.7% accuracy, respectively.

Using the Extracted Opinion Words for Improving Baseline. By using the extracted opinion words as features we were able to improve over the baselines in some cases. The average ten-fold cross-validation results are reported in the Table 11.

The description of the feature representation is presented in the Table 10.

Extracting Opinion Targets. Nouns that often co-occur with opinion words (both positive and negative) are likely to be opinion targets [12].

A list of nouns from the frequent bigrams with opinion words is presented below (32 words, frequency threshold is 6, translation is presented):

place, actor, action, impression, time, life, play, idea, story, picture, cinema, comedy, end, place, moment, music, image, feeling, viewing, serial, tale, meaning, trick, scene, plot, theme, thriller, movie, part, man, joke, effect, humour

With the frequency threshold 5 the following words were obtained additionally to the nouns above (12 words, translation is presented):

version, character, book, love, cartoon, evaluation, character, job, episode, situation, side, sense

A list of 21 newly extracted nouns for the frequency threshold 4 is presented below:

[3] http://rapid-i.com/

Table 10. Feature representation models for improving the baselines: description

	Description
1.1, 1.2	bigrams *Noun* + *Adj*, which occured at least k times in positive and k times in negative reviews
2	1.1, 1.2 + unigrams with the frequency threshold 2, 2
3	opinion words, obtained with the conditional probability based method with the frequency threshold 3, 3, reviews are represented as vektors
4	3 + unigrams with the frequency threshold 2, 2
5.1	Feature representation 2, described in 4.2; 9 opinion targets are manually created: *“movie”, “acting”, “actor”, “director”, “comedy”, “melodrama”, “cartoon”, “horror movie”, “plot”*; opinion words, obtained with the conditional probability based method with the frequency threshold 3, 3
5.2–5.4	similar to 5.1, but *OT* list of the length k is automatically created (see “Extracting Opinion Targets”)
6.1–6.2	Feature representation 2, described in 4.2 and unigrams
7	300 terms with the highest and 300 terms with the lowest *dev*, described in 3.4
8	7 + unigrams
9	8 + Feature representation 2, described in 4.2
10	9 + 2 features: “!” and difference between the number of “)” and “(”

Table 11. Feature representation models for improving the baselines: results

	Features	# feat.	Fr./Pr.	KNN	NB	SVM	LR
1.1	$N + Adj(1,1)$	23	Pres.	60.2	59.0	62.2	60.3
1.2	$N + Adj(0,0)$	42	Pres.	65.7	67.2	68.5	65.3
2	$N + Adj(0,0)$, $un._{(2,2)}$	783	Pres.	95.2	72.7	86.5	95.0
3	$OW_{(3,3)}$	1544	Pres.	92.4	73.2	86.3	90.7
4	$OW_{(3,3)}$, $un._{(2,2)}$	2285	Pres.	**95.7**	73.2	87.1	95.0
5.1	$Repr_2\ OT^m_9\ OW_{(3,3)}$	20	Pres.	50.6	63.2	68.5	68.1
5.2	$Repr_2\ OT^a_{45}\ OW_{(3,3)}$	92	Pres.	57.4	59.1	71.3	71.3
5.3	$Repr_2\ OT^a_{90}\ OW_{(3,3)}$	182	Pres.	60.8	61.0	68.4	71.9
5.4	$Repr_2\ OT^a_{385}\ OW_{(3,3)}$	772	Pres.	63.8	63.0	69.1	73.1
6.1	$Repr_2\ OT^m_9\ OW_{(3,3)}$, $un._{(2,2)}$	761	Pres.	94.6	73.0	87.0	94.9
6.2	$Repr_2\ OT^a_{385}\ OW_{(3,3)}$, $un._{(2,2)}$	1126	Pres.	94.3	62.8	85.7	**95.2**
7	dev (Top^{300}_{min}, Top^{300}_{max})	600	Pres.	74.4	56.7	73.1	73.3
8	dev (Top^{300}_{min}, Top^{300}_{max}), $un._{(2,2)}$	1341	Pres.	94.8	76.4	86.4	94.7
9	dev, $Repr_2\ OT^a_{385}$, $un._{(2,2)}$, $OW_{(3,3)}$	3657	Pres	**95.6**	80.6	88.6	**95.9**
10	9, Signs	3659	Pres	**95.8**	80.6	88.9	**96.1**

auditorium, variant, age, question, eye, graph, drama, viewer, quantity, thought, plan, advantage, continuation, role, word, shooting, level, outcome, worth, emotion, event

Feature Representation. In the current section two types of the object-attribute matrix for machine learning are considered.

Representation 1. Attributes are opinion words

Representation 2. An object-attribute matrix is created for machine learning according to the given in Table 12.

Table 12. An object-attribute matrix

	OT^+	OT^-	OT_1^+	OT_1^-	OT_2^+	OT_2^-	...	OT_n^+	OT_n^-
Doc_1	a_{10}^+	a_{10}^-	a_{11}^+	a_{11}^-	a_{12}^+	a_{12}^-	...	a_{1n}^+	a_{1n}^-
Doc_2	a_{20}^+	a_{20}^-	a_{21}^+	a_{21}^-	a_{22}^+	a_{22}^-	...	a_{2n}^+	a_{2n}^-
...	...	...	...	...	...	...	...	...	...
Doc_m	a_{m0}^+	a_{m0}^-	a_{m1}^+	a_{m1}^-	a_{m2}^+	a_{m2}^-	...	a_{mn}^+	a_{mn}^-

Here OT^+ and OT^- are the number of positive and negative opinion words in the document Doc_i, OT_i^+ and OT_i^- are the number of occurrences of the opinion target OT_i in the positive and negative contexts respectively, where OT_i is the word from the list opinion targets. By the term "context of OT_i" we mean the surrounding radius of the word by 5 words. The context can be called positive (negative) if it contains a positive(negative) opinion word.

The best accuracy, 96.1%, was obtained using simultaneously all opinion words extracted by all the four methods together with unigrams and features involving opinion targets.

5 Conclusion and Future Work

In this paper, we considered several methods for automatic extraction of domain-specific opinion words. For one of these methods, double propagation [13], we proposed a new technique for building a seed to be used as a starting point for opinion word extraction. Syntactic rules of the double propagation method were adapted for Russian.

We used several machine learning methods for determining text polarity. For this, we developed several text representations using various features and compared them empirically. Our experiments showed that opinion words are important for determining the polarity.

Using all unigrams as the only features during classification, we were able to obtain the accuracy of 95.2%. Adding opinion words to the list of features made it possible to improve the accuracy up to 96.1%. The gain in accuracy may not seem large at first glance, but note that the classification error was reduced by almost 20%. However, our experiments showed that, although opinion words are useful for polarity detection, they are not sufficient on their own and should be used only in combination with other features. Our results are better than those described in [10] and [9], which report the accuracy of 82.9% and 89% respectively. It remains to see whether this difference is due to our choice of the dataset or if, perhaps, the polarity of texts in Russian is easier to determine than that of texts in English, at least for some genres.

Acknowledgments. The results of the project “Mathematical Models, Algorithms, and Software Tools for Intelligent Analysis of Structural and Textual Data”, carried out within the framework of the Basic Research Program at the National Research University Higher School of Economics in 2012, are presented in this work.

References

1. Chetviorkin, I., Loukachevitch, N.: Automatic extraction of domain-specific opinion words. In: Proceedings of “Dialogue” International Conference (2010) (in Russian)
2. Church, K.W., Hanks, P.: Word association norms, mutual information and lexicography. In: Proceedings of the 27th Annual Conference of the ACL (1989)
3. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of ACL 1997, Stroudsburg, PA, pp. 174–181 (1997)
4. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: Proceedings of SIGKDD 2004, Seattle, Washington, USA, pp. 168–177 (2004)
5. Kanayama, H., Nasukawa, T.: Fully automatic lexicon expansion for domain-oriented sentiment analysis. In: Proceedings of EMNLP 2006, pp. 355–363 (2006)
6. Kim, S.-M., Hovy, E.: Determining the sentiment of opinions. In: Proceedings of COLING 2004, pp. 1367–1373 (2004)
7. Kobayashi, N., Inui, K., Matsumoto, Y.: Extracting aspect-evaluation and aspect-of relations in opinion mining. In: Proceedings of EMLP 2007 (2007)
8. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer, Berlin (2011)
9. Mullen, T., Collier, N.: Sentiment analysis using support vector machines with diverse information sources. In: Proceedings of EMNLP 2004 (2004)
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of EMNLP 2002 (2002)
11. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Now–Publishers Inc., Hanover (2008)
12. Popescu, A.-M., Etzioni, O.: Extracting Product Features and Opinions from Reviews. In: Proceedings of EMNLP 2005 (2005)
13. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion Word Expansion and Target Extraction through Double Propagation. Computational Linguistics 37(1), 9.27 (2011)
14. Takamura, H., Takashi, I., Manabu, O.: Extracting semantic orientations of words using spin model. In: Proceedings of ACL 2005, pp. 133–140 (2005)
15. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning subjective language. Computational Linguistics 30(3), 277–308 (2004)
16. Zhang, L., Liu, B.: Identifying Noun Product Features that Imply Opinions. In: Proceedings of ACL 2011 (2011)

Measuring Feature Distributions in Sentiment Classification

Diego Uribe

Instituto Tecnológico de la Laguna
División de Posgrado e Investigación
Blvd. Revolución y Calz. Cuauhtémoc, Torreón, Coah., MX
diego@itlalaguna.edu.mx

Abstract. We address in this paper the adaptation problem in sentiment classification. As we know, available labeled data required by sentiment classifiers does not always exist. Given a set of labeled data from different domains and a collection of unlabeled data of the target domain, it would be interesting to determine which subset of those domains has a feature distribution similar to the target domain. In this way, in the absence of labeled data for a particular target domain, it would be plausible to make use of the labeled data corresponding to the most similar domains.

1 Introduction

A huge volume of opinionated text is nowadays a common scenario in the Web. It is precisely this enormous information that represents a great sandbox for developing practical applications for the industry and the government. For example, before buying a product or service, people look for someone else's experiences about the product on the web, i.e. opinions. The same occurs with services provided by government or industry. In this way, persons and organizations are interested in the use of opinionated text for decision optimization: persons taking care of their money, and organizations getting significant feedbacks for improving their products or services [2].

This increasing interest in the use of opinionated text demands automated opinion discovery and classification systems. Sentiment classification is basically a text classification task in which, instead of assigning one topic to the text, the attitude of the reviewer with respect to the text is determined. As emotions are not only described in many ways but also differently interpreted by persons, sentiment classification is one of the most defiant problems in computational linguistics.

To cope with the linguistic challenge demanded by the automatic classification of emotions, diverse machine learning techniques have been applied. Most of the current work has been concerned with the use of supervised classification techniques which require large amounts of labeled training data [12],[5]. However, as labeling data is time-consuming, in some domains only limited training data exists. As training data might be limited, it could be fruitful revealing to

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 69–80, 2013.

investigate how existing resources could be used to train a classifier in different domains.

Since sentiment in different domains can be expressed in very different ways, training a classifier using data from one domain may fail when testing against data from another, different domain. In other words, we face greater difficulty when the available training instances differ from the target domain. Aue and Gamon illustrate how the accuracy of a classifier trained on a different domain drops significantly compared to the performance of a classifier trained on its own native domain [1]. Thus, to determine which subset of outside domains has a feature distribution most similar to the target domain is of paramount importance.

In this paper, it is proposed a framework for estimating which subset of labeled corpora domains has a feature distribution similar to the unlabeled target domain. Our approach gets its inspiration from the task of identifying the author of a given document -authorship attribution [6]. To be more specific, given a set of labeled data from different domains, a domain's profile is created by information extracted from the reviews corresponding to each domain. Once the profiles have been defined, an unlabeled review, corresponding to the target domain, is associated with the domain whose profile is most similar to the review. Thus, the specification of the linguistic terms to be extracted from each review is crucial for the definition of each domain's profile as well as the similarity metric used to identify unseen reviews.

We evaluate the proposed method with a data collection of several domains [17]. In particular, we used data from 4 different domains: Books, Movies, Music, and Phones. The feature distribution for each of the held-out domains is generated using profiles created on data from the other three domains. We then verify the feature distribution with classifiers based on the profile of each of the three domains and tested on the held-out domain (target domain). The results of the experimentation show how a feature distribution based on the semantic orientation of available datasets is a plausible option for the classification of unlabeled reviews.

The rest of the paper is organized as follows. The next section makes a brief review of of related work on the problem of domain adaptation in sentiment analysis. Section 3 describes in detail the proposed method for the creation of each domains profile and the metrics for computing the similarity between unlabeled reviews and domains profiles. Then, the setup and results of the experimentation are exhibited and discussed in section 4. Finally, conclusions are given in section 5.

2 Related Work

There is substantial work in sentiment classification using different strategies. Indeed, diverse machine learning techniques have been applied to automatically classify opinions. In this paper we briefly describe some of the preceding works into customizing sentiment classifiers to new domains we briefly mention some interesting works. One of the first works in this specific topic was carried out by

Aue and Gamon [1]. Their work is based on multiple ways of fusion of labeled data from other domains and unlabeled data from the target domain. The best results were obtained with an approach based on bootstrapping techniques. It is important to notice our investigation was initially proposed in this previous work.

Yang et al. [20] also deal with domain adaptation in their opinion detection system. Subjective and objective sentences from movies and products reviews were used as the training data for the task of opinion detection across the blog posts. Multiple feature vectors based on n-grams were used in the experimentation. Only those features highly ranked in both domains were considered as generic features for the classification task. Likewise, Blitzer et al. [4] cope with the domain adaptation problem by extending an algorithm for sentiment classifier by making use of *pivot features* that relate the source and target domains. This relationship is defined in terms of frequency and mutual information estimation.

Shou-Shan et al. [14] propose an interesting algorithm for multi-domain sentiment classification based on the combination of learners for specific domains called *member classifiers*. The combination of these member classifiers is done according to two types of rules: fixed and trained rules. The purpose of the combination process is to obtain and to make available global information to the final classifier.

3 Proposed Method

In this section, we describe in detail our approach to estimate the subset of different domains with a feature distribution similar to the target domain. As we previously said, our approach gets inspiration from the task of identifying the author of a given document, that is, authorship attribution [16]. As Figure 1 shows, there are two key elements in the framework. First, the creation of each domains profile requires the definition of the linguistic terms to be extracted from each review as well as the associated information of each term, i.e. frequency, semantic orientation. Second, the similarity metric used to get a clearer view on how distant the unlabeled reviews are from each previously defined profile. We analyze in this work two different ways to define a domain profile: a domain profile based on the presence of the subjective terms among diverse domains, and the semantic orientation of the subjective terms used as an alternative way to portray the profile of a particular domain.

3.1 Domain Profile

The first step to estimate the similarity distribution among domains is the creation of each domain's profile. Thus, to determine the granularity of the linguistic terms to be extracted from each review is crucial for the definition of a profile. Most of the previous work in sentiment classification makes use of n-grams models for the representation of the reviews ([12], [13], [20]). Also, another common

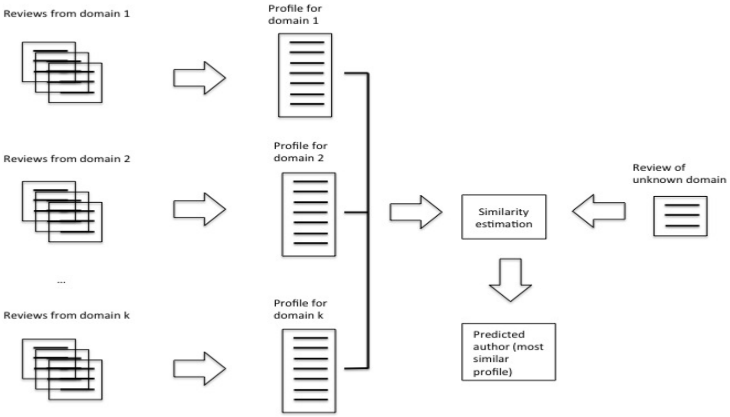


Fig. 1. Domain recognition

scenario is the use of linguistic patterns for the extraction of specific linguistic terms [19]. Since these patterns are based on adjectives and adverbs as good indicators of subjectivity and opinions, they represent a plausible alternative to look for an optimal subset of features. We use in this work a combined approach: a phrase pattern-based method which relies on bigrams and a small set of particular rules proposed by [15].

In fact, our method extracts bigrams as a way to incorporate elemental context in the analysis. But we don't extract any bigrams. We extract only those bigrams matched by the set of rules proposed by [15]. The key point in these rules is the consideration of the context that surrounds the subjective words [12]. For example, phrases such as *"nice"* and *"not very nice"* denote opposite emotions. Thus, the rules are basically applied to adjectives and adverbs and operate against only the three leading contextual word positions. It is also important to observe that the application of the rules is carried out in sequential order. Table 1 shows the rules along with some examples.

In this way, in order to obtain the subjective expressions matched by the linguistic patterns, the set of sentences corresponding to each review was submitted to a tagger: The Stanford Tagger [18]. Thus, a particular *domain profile* is the set of subjective bigrams extracted from their corresponding reviews.

To give a more formal definition, let d represents a specific domain. Then, a *domain profile (dp)* is defined as follows:

$$dp = \{t \mid t \text{ is a subjective term extracted from each } r \text{ where } r \in d\} \qquad (1)$$

3.2 Similarity Distribution

To estimate the similarity domain distribution a new unlabeled review and a domain profile are compared. A straightforward and common way to estimate the

Table 1. Steck's patterns

Rule	**Example**
Three leading positions preceding an adjective and attempts to match negations	*is not very **nice***
To match the first leading word that is not a preposition, conjunction, or punctuation mark	*movie was just **bizarre***
Applies when the trailing position is identified as a noun	***good** actor*

similarity between documents represented by vectors is based on the number of overlapping terms where the Dice, Cosine, and Jaccard coefficients are common metrics. Also, a recent interesting work on social bookmarking makes use of the coefficients to measure the similarity between users of portals for sharing scientific references amongst researchers [8]. However, we observe in this work from a different perspective the intersection of the linguistic terms between different domains and new unlabeled reviews. Two schemes are particularly analyzed: one is based on authorship attribution and the other one relies on semantic orientation.

Presence. In the base model of authorship attribution the use of normalized and un-normalized number of overlapping terms is commonly used as similarity measures [10]. However, we focus our attention on an interesting alternative to the base model for authorship identification suggested by Escalante [6]. Besides considering the number of overlapping terms, the basic idea is to assign a particular weight to each term according to their use across the profiles for different domains. This is precisely the approach that has been adapted in our work to estimate the similarity domain distribution.

Now, if we have k different domains, a domain profile, represented as dp_k, is defined for each particular domain. Then, the weight associated to each t common term between an unseen review and a particular domain profile is defined as follows:

$$w_t = \frac{1}{\sum_{i=1}^{k} at(t, dp_i)} \tag{2}$$

where $at(t, p_i)$ represents a *presence* function defined as:

$$at(t, dp_i) = \begin{cases} 1 & \text{if } t \in dp_i \\ \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

These expressions show how prolific terms among different domains are penalized with an irrelevant weight, whereas those terms whose occurrence is not so common among the profiles strongly influences the assignation of an unlabeled review to a particular domain. Thus, the similarity measure between a domain profile and an unseen review is defined as follows:

$$sim(r, dp_k) = \frac{1}{n} \times \sum_{i=1}^{n} w_i \tag{4}$$

where n represents the number of terms that overlap between the domain profile dp_k and the unseen r review. In other words:

$$n = r \cap dp_k \tag{5}$$

As we can see, the similarity value in (4) is not only obtained with the total of the common terms' weights. This total is normalized by n overlapping terms in order to reduce the impact of a review with a large number of subjective terms. Thus, the unseen r review will be assigned to the domain profile with the largest similarity value among k different domains.

Semantic Orientation. An alternative approach to estimate the similarity domain distribution is the incorporation of the semantic orientation of the linguistic elements used in the representation of the reviews. The classic Turney's work [19] makes use of a search engine to determine the semantic orientation of the extracted phrases by using PMI between the phrase and polarity words. However, since opinion lexicons are resources that associate words with sentiment orientation, the use of opinion lexicons is also another option for the estimation of the semantic orientation of the terms.

Basically, there are two alternatives to incorporate semantic lexicons in sentiment analysis. First, the use of SentiWordNet (SWN), an extension of WordNet with information about the sentiment of words where each synset has a positive, negative and objectivity score [7], is increasingly common in sentiment analysis. For example, Keefe makes use of SWN in order to propose feature selection methods to reduce the huge number of features found in opinion corpora [9]. Manually created opinion lexicons are another alternative. [12] uses this option with a group of students where they chose the polarity of the terms. In this way, the document's polarity is given by counting positive and negative terms.

In this work we don't make use of SWN and don't manually create a semantic lexicon either. In our phrase pattern-based method, we automatically construct two semantic lexicons for each domain to determine the semantic orientation of each subjective term: the positive and the negative semantic lexicons. The positive lexicon (PL) is automatically created with the whole set of extracted terms corresponding to the positives reviews. In a similar way, the negative semantic lexicon (NL) is also defined. Thus, the semantic orientation (so) of each particular subjective term (t) is estimated according to:

Let ρ be the number of times t occurs in PL

and η be the number of times t occurs in NL,

the degree of subjectivity is described as:

$$\mu = \rho - \eta \tag{6}$$

Then, the semantic orientation associated with each common term corresponding to a particular domain profile (dp) is defined as:

$$so(t, dp) = \begin{cases} 1 & \text{if } \mu > 0 \\ -1 & \text{if } \mu < 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Once the semantic orientation of a subjective term has been stated, we can now to determine the weight associated to each common term across different domain profiles. Assuming we have k different domains and its corresponding domain profile dp_k, the weight associated to each common term between a an unseen review and a particular domain profile is defined as follows:

$$w_t = \sum_{i=1}^{k} so(t, dp_i) \tag{8}$$

This expression shows how the weight of a term is determined by the subjectivity that such term entails across diverse domains rather than a specific domain. Terms with different polarity across domains (i.e. positive sentiment in one domain and negative sentiment in another) exhibit an irrelevant weight, whereas those terms with regular polarity among the profiles strongly influences the assignation of an unlabeled review to a particular domain. Thus, the similarity measure between a domain profile and an unseen review is defined as follows:

$$sim(r, dp_k) = |\sum_{i=1}^{n} w_i| \tag{9}$$

As we can see, the similarity value in (9) is not only obtained with the total of the common terms' weights. The absolute value of this total is used in order to emphasize the magnitude of the subjectivity that a review entails. Thus, the unseen r review will be assigned to the domain profile with the largest similarity value among k different domains.

As we can see, the similarity value in (9) is obtained with the total of the common terms' weights in order to emphasize the magnitude of the subjectivity that a review entails. Thus, the unseen r review will be assigned to the domain profile with the largest similarity value among k different domains.

4 Experimental Evaluation

We use for our experimental trial the collection of Epinions reviews[1] developed by Taboada et al. [17]. From the whole collection, four domains were selected:

[1] www.epinions.com

books, movies, music and phones. There are 50 opinions per category, and since there are 25 reviews per polarity within each category, the baseline accuracy for each domain is 50%. The set of sentences corresponding to each review was submitted to the Stanford Tagger [18] for the specification of the subjective terms to be extracted from each review.

4.1 Similarity Distributions Results

In order to observe the subset of domains with a feature distribution similar to the target domain, each of the selected domains play the role of target domain. For example, to obtain the feature distribution similar to Books, the set of opinions corresponding to Books play the role of unseen reviews to be identified across the Movies, Music, and Phones domains.

Two methods have been described in Section 3 to estimate such similarity distributions. The results of the first method based on the presence of the subjective expressions across the different domain's profiles are shown in Figure 2. The results are displayed in such a way that the column corresponding to each target domain is the lowest one. We observe for example how Music reviews exhibits a feature distribution more similar to Movies than Books.

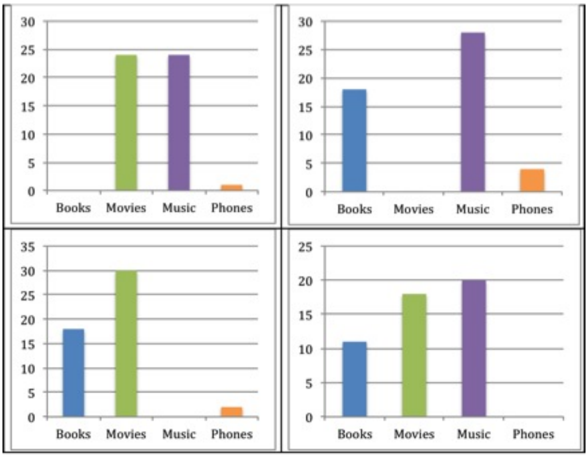


Fig. 2. Presence distribution

In the same way the results of the second method based on the semantic orientation of the subjective expressions across the different domains' profiles are shown in Figure 3. Also, the results are displayed in such a way that the column corresponding to each target domain is the lowest one. In this case we observe for example how Movies reviews exhibits a feature distribution more similar to Books than Music.

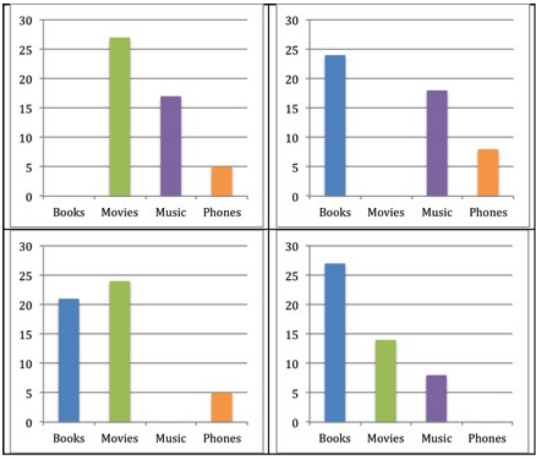


Fig. 3. Semantic orientation distribution

4.2 Analysis and Discussion

By observing the similarity distributions displayed in Figures 1 and 2, we can appreciate some coincidences and differences. For example, we can see how Movies show a feature distribution more similar to Books in both methods, even tough the results in the semantic orientation distribution are more evident. The same occurs when we obtain the distribution for Music: we can see how Movies show a feature distribution more similar to Music in both methods, but the results in the distribution based on presence are more palpable.

On the other hand, the distributions for Movies and Phones differ. For example, we can see how Music show a feature distribution more similar to Movies in the distribution based on presence, but the results in the semantic orientation distribution show Books as the most similar dataset. And the same occurs in Phones. We can see how Music show a feature distribution more similar to Phones in the distribution based on presence, but the results in the semantic orientation distribution show Books as the most similar dataset.

Once we obtained the similarity distributions for each target domain, we want to corroborate such distributions. In other words, we want to observe if, for example, Movies reviews represent a better option to classify Music rather than Books reviews. In order to carry out this corroboration, we evaluate the performance for each target domain using a classifier based on the common features between the unseen reviews of the target domain and each of the foreign domains. We implement two different methods to evaluate the performance.

The first experiment is based on 5-fold cross-validation and the use of support vector machines (SVM). As we know, SVM is a hyperplane classifier that has proved to be a useful approach to cope with natural text affairs [12]. Table 2 shows the SVM classifier accuracy for each target domain. Table 2 exhibits how the use of Movies reviews (60.00%) represent a better option to classify Music rather than Books reviews (56.00%). Taking into account that the baseline accuracy for each domain is 50%, the performance for unseen reviews from Music and Movies is poor, whereas for unseen reviews from Books and Phones a plausible performance is obtained.

Table 2. Scores obtained with SVM

	Books	**Movies**	**Music**	**Phones**
Books		69.39%	65.31%	61.22%
Movies	42.00%		48.00%	32.00%
Music	56.00%	60.00%		60.00%
Phones	79.60%	79.60%	77.60%	

The second evaluation method portrays more optimist results. This alternative evaluation is based on the use of the two semantic lexicons created for the definition of each domain profile with a different approach: instead of using such domain profiles to identify unseen reviews, we use them to classify the unseen reviews. In this way, we determine the semantic orientation of each subjective term to represent an unseen review as a feature set that denotes features intensity rather than features presence. The classification of the unseen review r is then defined in a similar way to the expression (9).

Table 3 shows the accuracy results of classifiers based on semantic orientation for each target domain. In this Table 3 we see for example how the use of Books reviews represents a better option to classify Movies rather than Music reviews. Likewise, by comparing Table 2 and Table 3, we see how the results obtained with classifiers based on the semantic orientation of the common features also represent a plausible alternative.

However, the most important point has been to observe how the information provided by the similarity distributions (Figure 2 and Figure 3) allows selecting domains with a feature distribution similar to the unlabeled target domain. To cope with the problem of domain adaptation is important to make a deep analysis of the characteristics of the unlabeled target domain. These provide relevant information not only for the selection of the best learning algorithm [3], but also for the selection of domains with a feature distribution similar to the target domain. In this way, the use of metalearning techniques by Shou-Shan et al. [14] can be enriched by knowledge such as the performance distribution of dissimilar domains to the target data.

Table 3. Scores obtained with SO

	Books	**Movies**	**Music**	**Phones**
Books		65.31%	59.18%	59.18%
Movies	72.00%		62.00%	56.00%
Music	68.00%	74.00%		52.00%
Phones	77.55%	51.02%	65.31%	

5 Conclusions and Future Work

In this paper, we approached the adaptation problem in sentiment classification by determining the subset of domains with a feature distribution similar to the unlabeled target domain. In particular, our approach intends to show how the estimation of similarity distribution among different domains provides useful information for the classification of unlabeled reviews. Our framework is based on the definition of domains' profiles as well as similarity metrics used to identify unseen reviews. The results show how a feature distribution based on the semantic orientation of available datasets is a plausible option for the classification of unlabeled reviews.

As part of our future work, we intend to explore the behavior of our method with different datasets. In particular, we are interested in the use of the dataset collected by Blitzer et al. [4]. This dataset is basically a collection of product reviews from four domains: books, DVDs, electronics, and kitchen appliances. Since each domain contains 1,000 positive and 1,000 negative reviews, we think this collection is worth our attention.

Finally, we also intend to deep in the exploration of a meta-level system to identify the most similar datasets for a given input dataset [3]. The purpose is to understand the connection between learning algorithms and the characteristics of the data under analysis.

References

1. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: A case study. In: Proceedings of Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria (2005)
2. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, 1st edn. Springer (2007)
3. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning Applications to Data Mining, 1st edn. Springer (2009)
4. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, Boom-boxes and Blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic (2007)
5. Chesley, P., Vincent, B., Xu, L., Srihari, R.: Using verbs and adjectives to automatically classify blog sentiment. In: AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW), pp. 74–80 (2006)
6. Escalante, H.J., Montes, M., Solorio, T.: A Weighted Profile Intersection Measure for Profile-based Authorship Attribution. In: Proceedings of 10th Mexican International Conference on Artificial Intelligence (MICAI), Puebla, Mexico (2011)
7. Esuli, A., Sebastiani, F.: SentiWordNet: a publicly available lexical resource for opinion mining. In: Proceedings of Fifth International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, vol. 6 (2006)
8. Heck, T.: A Comparison of Different User-Similarity Measures as Basis for Research and Scientific Cooperation. In: International Conference on Information Science and Social Media, bo/Turku, Finland (2011)

9. Keefe, T., Koprinska, I.: Feature Selection and Weighting Methods in Sentiment Analysis. In: Proceedings of the 14th Australasian Document Computing Symposium, Sydney, Australia (2009)
10. Keselj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the Pacific Association for Computational Linguistics, Halifax, Canada, pp. 255–264 (2003)
11. Liu, B.: Sentiment Analysis and Subjectivity. In: Indurkhya, N., Damerau, F.J. (eds.) Handbook of Natural Language Processing, pp. 627–665. Chapman and Hall/CRC (2010)
12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, USA, pp. 79–86 (2002)
13. Pang, B., Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Barcelona, Spain, pp. 21–26 (2004)
14. Shou-Shan, L., Chu-Ren, H., Cheng-Qing, Z.: Multi-Domain Sentiment Classification with Classifier Combination. Journal of Computer Science and Technology 26(1), 25–33 (2011)
15. Steck, J.B.: Netpix: A Method of Feature Selection Leading to Accurate Sentiment-Based Classification Models. Master thesis, Central Connecticut State University (2005)
16. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology 60(3), 538–556 (2009)
17. Taboada, M., Anthony, C., Voll, K.: Creating Semantic Orientation Dictionaries. In: Proceedings of Fifth International Conference on Language Resources and Evaluation (LREC), Genoa, Italy, pp. 427–432 (2006)
18. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), Edmonton, Canada, pp. 252–259 (2003)
19. Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, USA, pp. 417–424 (2002)
20. Yang, H., Si, L., Callan, J.: Knowledge transfer and opinion detection in the TREC2006 blog track. In: Proceedings of the Fifteenth Text REtrieval Conference, Gaithersburg, MD (2006)

Identification of the Minimal Set of Attributes That Maximizes the Information towards the Author of a Political Discourse: The Case of the Candidates in the Mexican Presidential Elections

Antonio Neme[1,2], Sergio Hernández[3], and Vicente Carrión[4]

[1] Complex Systems Group, Universidad Autónoma de la Ciudad de México
San Lorenzo 290, México, D.F. México
[2] Institute for Molecular Medicine, Finland
neme@nolineal.org.mx
[3] Postgraduation Program in Complex Systems, Universidad Autónoma de la Ciudad de México
[4] CINVESTAV IDS, México D.F.

Abstract. Authorship attribution has attracted the attention of the natural language processing and machine learning communities in the past few years. Here we are interested in finding a general measure of the style followed in the texts from the three main candidates in the Mexican presidential elections of 2012. We analyzed dozens of texts (discourses) from the three authors. We applied tools from the time series processing field and machine learning community in order to identify the overall attributes that define the writing style of the three authors. Several attributes and time series were extracted from each text. A novel methodology, based in mutual information, was applied on those time series and attributes to explore the relevance of each attribute to linearly separate the texts accordingly to their authorship. We show that less than 20 variables are enough to identify, by means of a linear recognizer, the authorship of a text from within one of the three considered authors.

Keywords: authorship attribution, mutual information, genetic algorithms.

1 Introduction

Authorship attribution (AA) and stylistics have attracted the attention of different practitioners from areas as diverse as computer science, philosophy, the arts, mathematics, and engineering. AA refers to the task of identifying the author of a text from a group of possible candidate authors [1], whereas stylistics refer to the identification of attributes that may lead to unequivocally identify an author [4]. Both tasks have witnessed an outstanding advance in recent years. However, there are still a lot of open questions.

One of the open questions is the identification of the minimum set of attributes that can lead to the identification of the author. Several attributes have been

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 81–90, 2013.

proposed, for example, the use of certain words and the lack of use of other [1]. In general, the concept of *bag of words* is frequently mentioned and, although relevant results have emerged, there are even more questions to be answered [2]. Writers use language following different ways to express their ideas. This variation in language allows the authorship attribution possible [3].

Several algorithms are able to identify the author of a given text, but, however, most of them lack of explanatory properties. For example, some kernel methods present good performance, but the model is unable to show what attributes are really relevant.

In this contribution, we present results regarding the identification of the minimal set of attributes that define a specific feature space. We are interested in such a feature space in which the mutual information between the coordinates of the points (texts) and the label (author of the text) is maximal. We present results regarding three authors, that happen to be the main candidates for the Mexican presidential elections of 2012. We selected the texts from several political discourses of those authors for two main reasons. The first one is that the subject in all texts for the three authors is very similar and is mainly in the economic public services, and taxes themes. This allows an easier isolation of the stylistics as it is not affected by a large variety of themes. The second reason is that of there are several texts available from candidates. Finally, it is relevant to know at least some aspect of stylistics from political leaders.

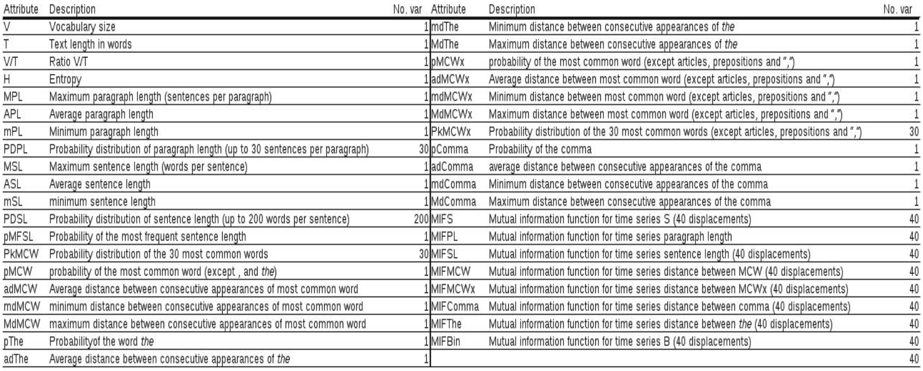
The rest of this contribution is presented as follows. In section 2 we describe the attributes that will lead to definition of the stylistics of the authors. In section 3 we present our proposal to identify the minimum set of attributes that define a space such that the coordinates of texts in that space give the maximum information about the author (class). In section 4 several results are described, and in section 5 some conclusions are discussed.

2 Attributes and Stylistics

Several attributes have been proposed as marks in order to discern the stylistics of an author [1]. Also, many features have been proposed to be relevant for the AA task, as vocabulary size or the use of certain words or structures [2]. Here we will focus our attention in attributes about the way authors make use of words. We refer to words as the vocabulary but also to punctuation marks.

In this contribution, we consider two kinds of attributes that are extracted from each text. The first one consists of probabilities of appearance of certain words. The second group consists of the mutual information of several time series constructed from texts. Fig. 1 show those attributes and an identification name.

Texts are represented as sequences of symbols, so they are transformed to series of integers. From the vocabulary for each text, each word is assigned an integer in order of appearance. The first word to appear in the text will be assigned to 0, the second non-repeated word 1, and so on. For example, the sentence $S =$ *In the city as well as in each neighborhood...* is transformed to the sequence $T = \{0, 1, 2, 3, 4, 3, 0, 5, 6, ...\}$. The word *in* is assigned to code 0 as it



Attribute	Description	No. var	Attribute	Description	No. var
V	Vocabulary size	1	mdThe	Minimum distance between consecutive appearances of *the*	1
T	Text length in words	1	MdThe	Maximum distance between consecutive appearances of *the*	1
V/T	Ratio V/T	1	pMCWx	probability of the most common word (except articles, prepositions and “,”)	1
H	Entropy	1	adMCWx	Average distance between most common word (except articles, prepositions and “,”)	1
MPL	Maximum paragraph length (sentences per paragraph)	1	mdMCWx	Minimum distance between most common word (except articles, prepositions and “,”)	1
APL	Average paragraph length	1	MdMCWx	Maximum distance between most common word (except articles, prepositions and “,”)	1
mPL	Minimum paragraph length	1	PkMCWx	Probability distribution of the 30 most common words (except articles, prepositions and “,”)	30
PDPL	Probability distribution of paragraph length (up to 30 sentences per paragraph)	30	pComma	Probability of the comma	1
MSL	Maximum sentence length (words per sentence)	1	adComma	average distance between consecutive appearances of the comma	1
ASL	Average sentence length	1	mdComma	Minimum distance between consecutive appearances of the comma	1
mSL	minimum sentence length	1	MdComma	Maximum distance between consecutive appearances of the comma	1
PDSL	Probability distribution of sentence length (up to 200 words per sentence)	200	MIFS	Mutual information function for time series S (40 displacements)	40
pMFSL	Probability of the most frequent sentence length	1	MIFPL	Mutual information function for time series paragraph length	40
PkMCW	Probability distribution of the 30 most common words	30	MIFSL	Mutual information function for time series sentence length (40 displacements)	40
pMCW	probability of the most common word (except , and *the*)	1	MIFMCW	Mutual information function for time series distance between MCW (40 displacements)	40
adMCW	Average distance between consecutive appearances of most common word	1	MIFMCWx	Mutual information function for time series distance between MCWx (40 displacements)	40
mdMCW	minimum distance between consecutive appearances of most common word	1	MIFComma	Mutual information function for time series distance between comma (40 displacements)	40
MdMCW	maximum distance between consecutive appearances of most common word	1	MIFThe	Mutual information function for time series distance between *the* (40 displacements)	40
pThe	Probabilityof the word *the*	1	MIFBin	Mutual information function for time series B (40 displacements)	40
adThe	Average distance between consecutive appearances of *the*	1			40

Fig. 1. The included attributes. Some are scalars, some are probability distributions, and others are mutual information functions (see next section).

is the first word. The second appearance of *in* is also assigned code 0. In this contribution, there is no difference between upper and lower cases.

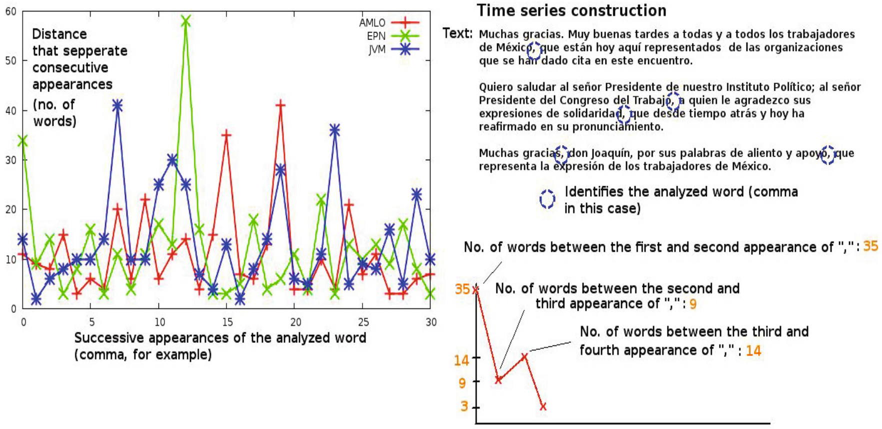
Time series from texts are relevant in the vision about stylistics we follow. Several time series were constructed from each text, and they are generated measuring the distance (counting the number of words) between consecutive appearances of certain words. For example, a certain time series that measures the distance (number of words) between consecutive appearances of the comma may reads as $\{3, 9, 40, 11\}$, which means that the number of words between the second and first appearance is 3, the distance between the second and third appearances is 9, and so on. Fig. 2 shows an example of the construction of the time series for the comma. Time series for the following instances were constructed:

- the comma
- sentence length (number of words between them)
- number of sentences per paragraph
- the most common word excluding the comma and the word *the*
- the most common word excluding articles and prepositions
- the word *the*

However, time series *per se* only give some visual details, and more processing on them is necessary.

Texts may present different lengths, so a normalizing scheme is needed in order to compare time series. At the same time, time series are not analyzed directly. In general several tools from the time series and signal processing fields can be applied in order to extract subtle and non-evident patterns [5]. Several attributes can be extracted from time series, such as the power spectrum, the Lyapunov exponent, and many others [6]. In this contribution, we applied mutual information function (MIF).

MIF is an information measure. Once we know the state of a system, How much information does that give about the state of which a second system?



Muchas gracias. Muy buenas tardes a todas y a todos los trabajadores de México, que están hoy aquí representados de las organizaciones que se han dado cita en este encuentro.

Quiero saludar al señor Presidente de nuestro Instituto Político; al señor Presidente del Congreso del Trabajo, a quien le agradezco sus expresiones de solidaridad, que desde tiempo atrás y hoy ha reafirmado en su pronunciamiento.

Muchas gracias, don Joaquín, por sus palabras de aliento y apoyo, que representa la expresión de los trabajadores de México.

Identifies the analyzed word (comma
in this case)
No. of words between the first and second appearance of ",": 35
35
No. of words between the second and
third appearance of ",": 9
No. of words between the third and
fourth appearance of ",": 14
14
9
3

Fig. 2. Time series construction. It is shown the case for distance (number of words) between consecutive appearances of comma. Only the first 31 appearances are shown.

MIF is based in Shannon's information theory. It is based on entropic-related concepts. 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 corresponds to the number of authors and 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 [8]. Mutual information between two random variables X and Z is expressed as $\varPhi(X;Z)$, 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:

$$\varPhi(X;Z) = \sum_{i}^{ns} \sum_{j}^{\#statesinZ} P(i,j) log \frac{P(i,j)}{P(i)P(j)} \tag{2}$$

The number of states in Z is 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. $P(i)$ is the marginal probability that a randomly chosen text belongs to a certain state, $P(j)$ is the marginal probability that the text belongs to class (author) j, and $P(i,j)$ is the joint probability that the text is in state i and belongs to author j. In general, for artificial datasets with no noise, all entropy in the label (class) can be removed from the list of attributes $\bar{X}$ that define the high-dimensional feature space. That is, $\varPhi(\bar{X};Z) = H$. Mutual information between the compound system of all attributes or variables and Z ($\varPhi(\bar{X};Z)$) tends to dissipate all entropy in the label. That is, when $ns \to \infty$, $\varPhi(\bar{X};Z) \to H$.

When the correlation between two systems (or random variables) is the mutual information, we are considering high-order momentum able to capture non-linear correlations in data [10,8].

When MIF is applied to a time series, the second system (or random variable) is constructed as a shift applied to the time series. The length of that shift is shown in the x axis. The graph of MIF then responds the question of how much information is achieved once we know the state of a system (the time series) with respect to the next state it will present (k = 1), two steps ahead (k = 2), and so on. Fig. 3

We call the whole set of attributes T

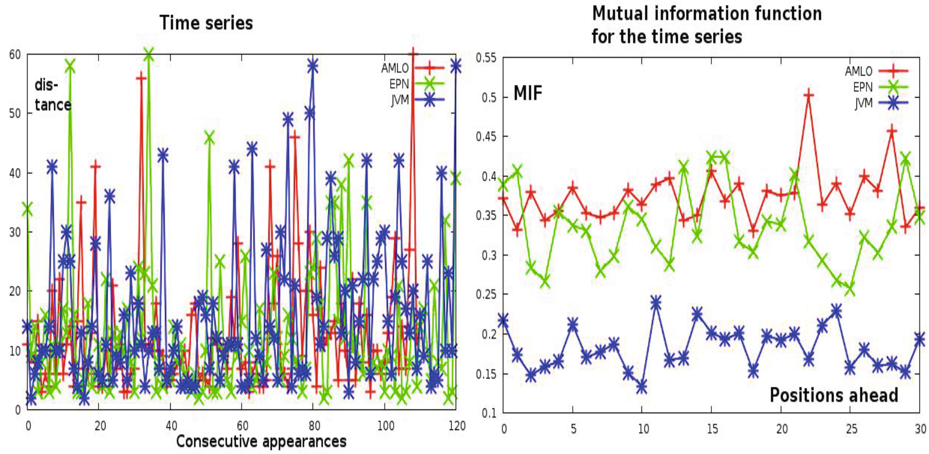


Fig. 3. Mutual information function (MIF) of the distance between commas time series. One time series for each one of the considered authors is shown.

3 The Proposed Model

Each text is transformed to a point in a high-dimensional space. The coordinates of each text are determined by the attributes described in the previous section. Points in this space may be the input data to a classification machine like multilayer perceptrons, identification trees or support vector machines so that a mapping between the coordinates and the label (the author) is found. If the number of attributes, that is, the dimension of the feature space, is high then the procedure followed to find such mapping, known as training process, may be very time consuming. In other cases, as in SVM, the new generated very-high-dimensional space lack explanatory power, that is, it may not be clear what attributes are really relevant in the classification task. On the other hand, classifiers based in trees such as C4.5 [12] offer an explicit explanation about the classification task. C4.5 and related methods, although relevant and useful in many situations, suffer from a major drawback: their greedy strategy may lead them to local optimum.

We are interested in finding a subset $A \in T$ such that $|A| \leq K$ such that the mutual information from A to the classes (author's name) is maximal. Let $\Phi(X, Y)$ be the mutual information between systems X and Y, and Note that this task is not equivalent to that performed by C4.5 related algorithms. We are not interested in classification by means of mutual information. We are trying to find a subspace such that coordinates in that space give as much information about the label or class as possible, and any machine learning algorithm can be fed with vectors in that space A, instead of being fed with vectors from space T whose dimensionality is higher. In that sense, our task is slightly similar to that of testors [13], in which a matrix of differences is systematically explored to identify those features that correctly classify patterns.

Once again, we intend to find an attribute or feature space such that the mutual information between points representing texts in that space and authors is as maximum as possible. In order to do this, the mutual information of a compound system is needed. That is, if there is only one attribute then the MI (mutual information) is calculated straightforward. In the case of two continuous attributes $X^{'}$ and $Y^{'}$ and ns is the number of states in which each attribute is to be discretized (X and Y), a compound system Z is constructed as follows. $Z'_i = X_i \times ns + Y_i$ and $Z = discretize(Z^{'}, ns)$. For more than two attributes, the procedure is applied recursively.

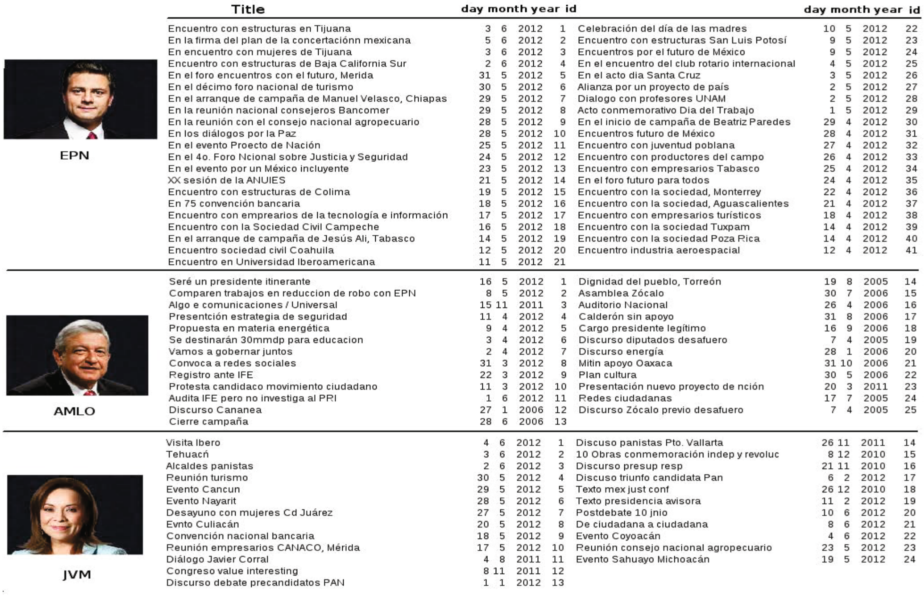
4 Results

The analyzed texts are shown in fig 4, along with the date they were dictated as a public discourse and a short title. All texts were preprocessed to remove transcribed messages from the public and other irrelevant information.

Fig. 5 shows the vocabulary as a function of number of words for the analyzed texts (see fig. 4). It is observed that, although one of the authors present a significant lower vocabulary size, it is still an attribute unable to give a lot of information about the author.

Fig. 6 shows the mutual information between some individual attributes in T and the class (author of the text). The dimension of space T, that is the number of attributes, is $D \sim 500$. That would require a lot of effot to train a multilayer perceptron. For a SVM, that may be an easy task, but we are interested in the identification of attributes that give information about the class. The function that maps from that space to the label space is not explored in this contribution. SVM does not offer such explanation.

The náive scheme to construct the space A from T will be to select the K most informative variables. Such strategy is followed, for example, by C4.5 and related algorithms, but that greedy strategy leads to local optima. In fig. 6 can be see an example of the failure of such strategy. If attributes V and T were rejected and H and $ANPL$ (see fig. 1) were selected, the opportunity that the compound system V, T to be selected would not be explored, and, as can



	Title	day	month	year	id		day	month	year	id
EPN	Encuentro con estructuras en Tijuana	3	6	2012	1	Celebración del día de las madres	10	5	2012	22
	En la firma del plan de la concertaciónn mexicana	5	6	2012	2	Encuentro con estructuras San Luis Potosí	9	5	2012	23
	En encuentro con mujeres de Tijuana	3	6	2012	3	Encuentros por el futuro de México	9	5	2012	24
	Encuentro con estructuras de Baja California Sur	2	6	2012	4	En el encuentro del club rotario internacional	4	5	2012	25
	En el foro encuentros con el futuro, Merida	31	5	2012	5	En el acto dia Santa Cruz	3	5	2012	26
	En el décimo foro nacional de turismo	30	5	2012	6	Alianza por un proyecto de país	2	5	2012	27
	En el arranque de campaña de Manuel Velasco, Chiapas	29	5	2012	7	Dialogo con profesores UNAM	2	5	2012	28
	En la reunión nacional consejeros Bancomer	29	5	2012	8	Acto conmemorativo Dia del Trabajo	1	5	2012	29
	En la reunión con el consejo nacional agropecuario	28	5	2012	9	En el inicio de campaña de Beatriz Paredes	29	4	2012	30
	En los diálogos por la Paz	28	5	2012	10	Encuentros futuro de México	28	4	2012	31
	En el evento Proecto de Nación	25	5	2012	11	Encuentro con juventud poblana	27	4	2012	32
	En el 4o. Foro Ncional sobre Justicia y Seguridad	24	5	2012	12	Encuentro con productores del campo	26	4	2012	33
	En el evento por un México incluyente	23	5	2012	13	Encuentro con empresarios Tabasco	25	4	2012	34
	XX sesión de la ANUIES	21	5	2012	14	En el foro futuro para todos	24	4	2012	35
	Encuentro con estructuras de Colima	19	5	2012	15	Encuentro con la sociedad, Monterrey	22	4	2012	36
	En 75 convención bancaria	18	5	2012	16	Encuentro con la sociedad, Aguascalientes	21	4	2012	37
	Encuentro con emprearios de la tecnología e información	17	5	2012	17	Encuentro con empresarios turísticos	18	4	2012	38
	Encuentro con la Sociedad Civil Campeche	16	5	2012	18	Encuentro con la sociedad Tuxpam	14	4	2012	39
	En el arranque de campaña de Jesús Ali, Tabasco	14	5	2012	19	Encuentro con la sociedad Poza Rica	14	4	2012	40
	Encuentro sociedad civil Coahuila	12	5	2012	20	Encuentro industria aeroespacial	12	4	2012	41
	Encuentro en Universidad Iberoamericana	11	5	2012	21					
AMLO	Seré un presidente itinerante	16	5	2012	1	Dignidad del pueblo, Torreón	19	8	2005	14
	Comparen trabajos en reduccion de robo con EPN	8	5	2012	2	Asamblea Zócalo	30	7	2006	15
	Algo e comunicaciones / Universal	15	11	2011	3	Auditorio Nacional	26	4	2006	16
	Presentción estrategia de seguridad	11	4	2012	4	Calderón sin apoyo	31	8	2006	17
	Propuesta en materia energética	9	4	2012	5	Cargo presidente legítimo	16	9	2006	18
	Se destinarán 30mmdp para educacion	3	4	2012	6	Discurso diputados desafuero	7	4	2005	19
	Vamos a gobernar juntos	2	4	2012	7	Discurso energía	28	1	2006	20
	Convoca a redes sociales	31	3	2012	8	Mitin apoyo Oaxaca	31	10	2006	21
	Registro ante IFE	22	3	2012	9	Plan cultura	30	5	2006	22
	Protesta candidaco movimiento ciudadano	11	3	2012	10	Presentación nuevo proyecto de nción	20	3	2011	23
	Audita IFE pero no investiga al PRI	1	6	2012	11	Redes ciudadanas	17	7	2005	24
	Discurso Cananea	27	1	2006	12	Discurso Zócalo previo desafuero	7	4	2005	25
	Cierre campaña	28	6	2006	13					
JVM	Visita Ibero	4	6	2012	1	Discuso panistas Pto. Vallarta	26	11	2011	14
	Tehuacń	3	6	2012	2	10 Obras conmemoración indep y revoluc	8	12	2010	15
	Alcaldes panistas	2	6	2012	3	Discurso presup resp	21	11	2010	16
	Reunión turismo	30	5	2012	4	Discuso triunfo candidata Pan	6	2	2012	17
	Evento Cancun	29	5	2012	5	Texto mex just conf	26	12	2010	18
	Evento Nayarit	28	5	2012	6	Texto presidencia avisora	11	2	2012	19
	Desayuno con mujeres Cd Juárez	27	5	2012	7	Postdebate 10 jnio	10	6	2012	20
	Evnto Culiacán	20	5	2012	8	De ciudadana a ciudadana	8	6	2012	21
	Convención nacional bancaria	18	5	2012	9	Evento Coyoacán	4	6	2012	22
	Reunión empresarios CANACO, Mérida	17	5	2012	10	Reunión consejo nacional agropecuario	23	5	2012	23
	Diálogo Javier Corral	4	8	2011	11	Evento Sahuayo Michoacán	19	5	2012	24
	Congreso value interesting	8	11	2011	12					
	Discurso debate precandidatos PAN	1	1	2012	13					

Fig. 4. The texts considered in this contribution. The authors are the three main candidates for Mexican presidential elections in 2012. 41 texts were selected for author EPN, 25 for author AMLO and 24 for author JVM. The selection criteria was that text should be within a certain range of number of words.

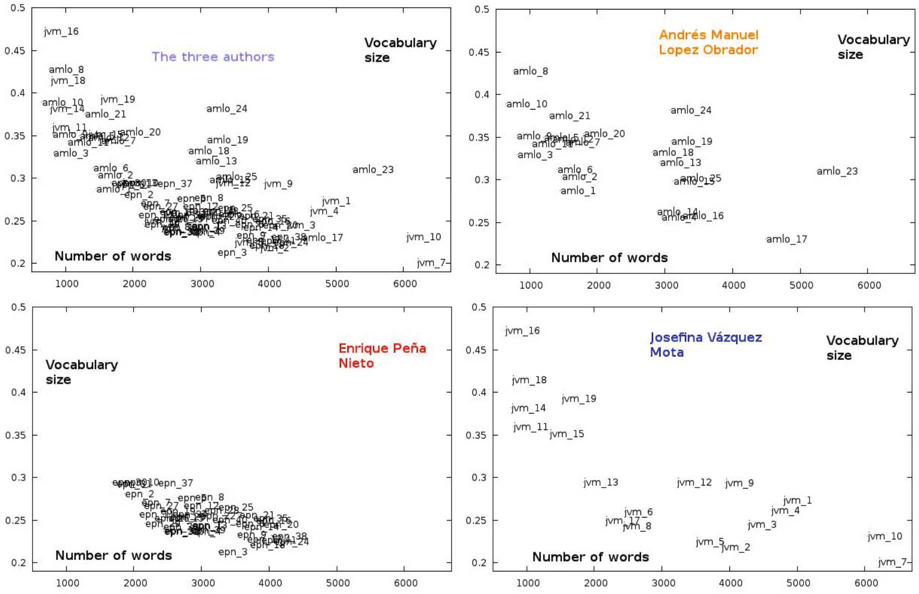


Fig. 5. Vocabulary size as a function of text length (number of words)

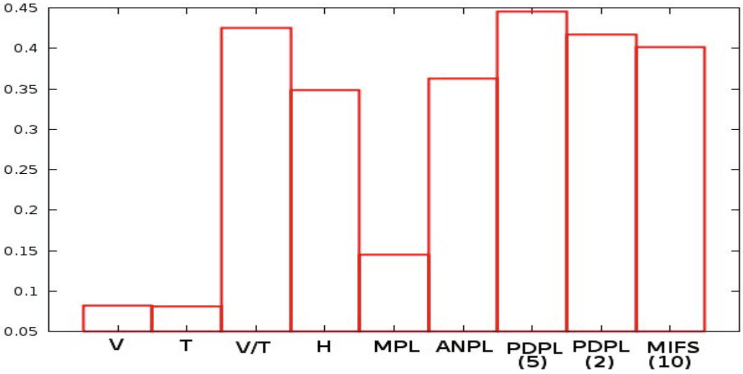


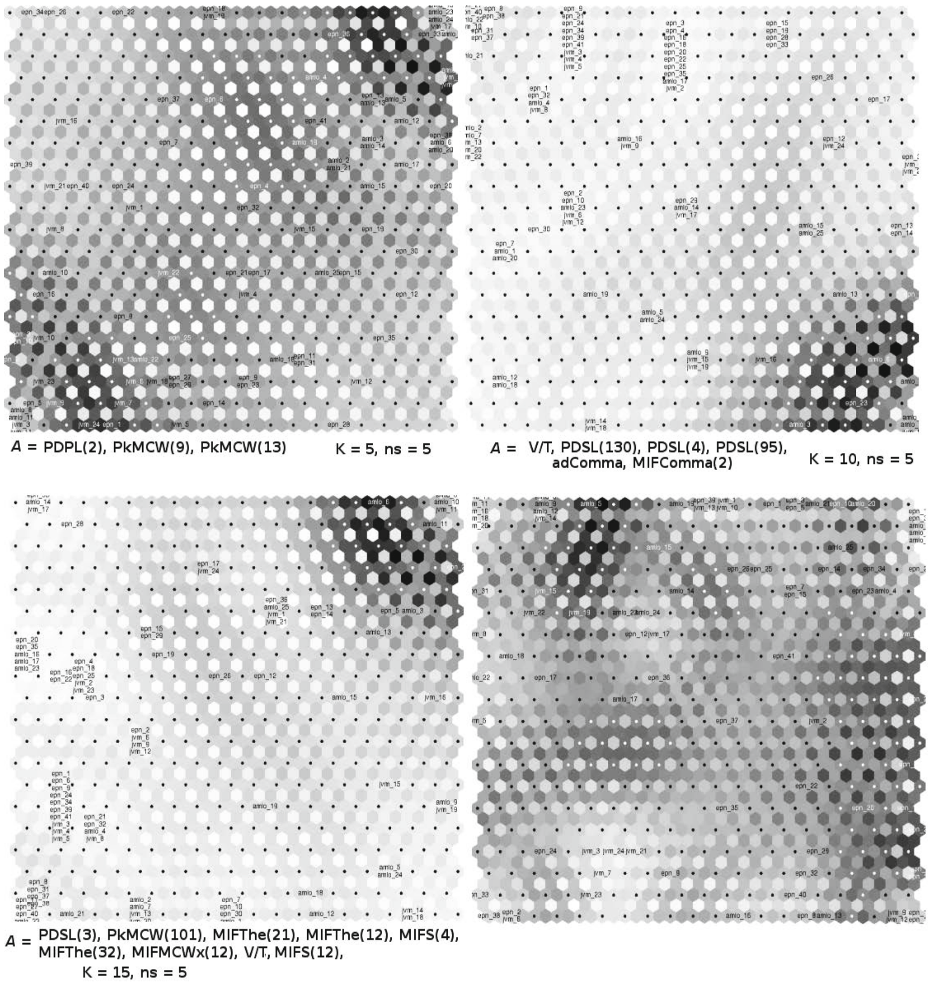
Fig. 6. Mutual information between some of the attributes listed in fig. 1 and the text author (class)

be seen, a compound system (in this case formed as the ratio) was in fact a better option. Thus, we want to explore space T instead of disregarding some of the variables from the beginning. For probability distributions and mutual information functions, the number in the parenthesis represent the component. For example $PDSL(4)$ refers to the probability of sentences of 4 words.

The space generated by K attributes from space T is called A. The number of possible spaces A is the number of permutations of K positions available to D different attributes $C(D, K)$. The exhaustive search for the case here presented is prohibitively time consuming for $K > 3$. Thus, a search scheme is needed. We applied an heuristic search method, the genetic algorithm, in order to find at most K attributes from T that generate a space such that $\Phi(A; Class)$ is maximum.

We implemented a genetic algorithm in Python, with elitism and probabilities of mutation of 0.05 and crossover of 0.9. Population size was settled to 100 and the algorithm was allowed to run for 500 epochs.

Note that the algorithm identifies a space A of dimension $D \leq K$. That space is not easily observed once $D > 3$. In order to visualize the distribution of the analyzed texts in that space, a mapping algorithm is needed. We decanted our options towards the self-organizing map (SOM) as it is a powerful visualization tool. SOM is able to present in a low-dimensional space an approximate distribution that resembles the actual distribution of vectors in the high-dimensional input space [14]. It outperforms common mapping tools such as principal component analysis as SOM is able to account for high-order statistics, instead of at most second-order (variance) [15]. In fig. 7 several maps obtained by SOM are presented. It can be observed that, indeed, there are detectable general distribution patterns that may allow to discriminate the author. Texts do not necessarily form clusters: once again, we are interested in an attribute space such that mutual information between the distribution and the author of a text is maximized. Clusters are only one way in which that mutual information can be maximized, but there are many others. Our methodology finds a family of those distributions.



A = PDPL(2), PkMCW(9), PkMCW(13) K = 5, ns = 5

A = V/T, PDSL(130), PDSL(4), PDSL(95), adComma, MIFComma(2) K = 10, ns = 5

A = PDSL(3), PkMCW(101), MIFThe(21), MIFThe(12), MIFS(4), MIFThe(32), MIFMCWx(12), V/T, MIFS(12), K = 15, ns = 5

Fig. 7. SOM (low-dimensional approximations) of feature spaces A

In fig. 7, besides the self-organizing map, it is shown the space A. Now, a machine with universal approximation capabilities such as the multilayer perceptron can be applied to space A, instead of being fed by data in space T. The fact that data can be explored in order to identify the combination (subspaces) of attributes that offer the maximum mutual information decreases the training time. Also, it points to the most relevant joint variables, which are in A. It may be possible to add new variables to A as the mutual information will not decrease [11]. However, eliminating attributes from A may have dramatic consequences, as the removed variable may be of great relevance.

5 Conclusions

In the tasks of authorship attribution and computational stylistics, it is of major interest to identify a set of attributes that can offer as much information as possible about the author of the text. Here, we have systematically explored schemes for detecting a subset of a large number of variables that can maximize information about the author. A genetic algorithm that constructs a space of at most K attributes such that it maximized the information about the class or author of the text was implemented.

The methodology here described can be applied to any kinds of texts. Here, we reported results for a special case, regarding political discourses, but in our project (not enunciated for double blind review purposes) we are applying these and other methodologies in order to study computational stylistics and style evolution.

References

1. Juola, P.: Authorship attribution. NOW Press (2008)
2. Stamatatos, E.: A survey of modern authorship attribution methods. J. of the American Soc. for Information Science and Technology 60(3), 538–556 (2010)
3. Neme, A., Cervera, A., Lugo, T.: Authorship attribution as a case of anomaly detection: A neural network model. Int. J of Hybrid Intell. Syst. 8, 225–235 (2011)
4. Manning, C., Schutze, H.: Foundations of statistical natural language processig. MIT Press (2003)
5. Abarbanel, H.: Analysis of observed chaotic data. Springer (1996)
6. Kantz, H., Schreiber, T.: Nonlinear time series analysis, 2nd edn. Cambridge Press
7. Shannon, C.E.: A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
8. Cellucci, C.J., Albano, A.M., College, B., Rapp, P.E.: Statistical Validation of Mutual Information Calculations. Phy. Rev E. 71(6) (2005), 10.1103/PhysRevE.71.066208
9. Santos, J., Marques de Sá, J., Alexandre, L., Sereno, F.: Optimization of the error entropy minimization algorithm for neural network classification. In: ANNIE V. 14 of Intelligent Engineering Systems Through Art. Neural Net, pp. 81–86. ASME Press, USA (2004)
10. Silva, L., Marques de Sá, J., Alexandre, L.: Neural Network Classification using Shannon's Entropy. In: ESANN 13th European Symp. on Art. Neural Net (2005)
11. Cover, T., Thomas, J.: Elements of information theory, 2nd edn. Wiley (2006)
12. Quinlan, R.: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)
13. Cortes, M.L., Ruiz-Shulcloper, J., Alba-Cabrera, E.: An overview of the evolution of the concept of testor. Pattern Recognition 34, 753–762 (2001)
14. Kohonen, T.: Self-organizing maps, 2nd edn. Springer (2000)
15. The Self-Organizing Maps: Background, Theories, Extensions and Applications. Studies in Computational Intelligence (SCI), vol. 115, pp. 715–762 (2008)

Document Categorization Based on Minimum Loss of Reconstruction Information

Juan Carlos Gomez and Marie-Francine Moens

Katholieke Universiteit Leuven
Department of Computer Science, Celestijnenlaan 200A, B-3001 Heverlee, Belgium
{juancarlos.gomez,sien.moens}@cs.kuleuven.be

Abstract. In this paper we present and validate a novel approach for single-label multi-class document categorization. The proposed categorization approach relies on the statistical property of Principal Component Analysis (PCA), which minimizes the reconstruction error of the training documents used to compute a low-rank category transformation matrix. This matrix allows projecting the original training documents from a given category to a new low-rank space and then optimally reconstructs them to the original space with a minimum loss of information. The proposed method, called Minimum Loss of Reconstruction Information (mLRI) classifier, uses this property, extends and applies it to unseen documents. Several experiments on three well-known multi-class datasets for text categorization are conducted in order to highlight the stable and generally better performance of the proposed approach in comparison with other popular categorization methods.

1 Introduction

Document categorization plays an important role in the flow of document interchanges, since it facilitates the tasks of accessing and retrieving relevant information by users and systems. Document categorization is a key component for many practical applications such as digital libraries and Web search engines.

In this article we are interested in the text categorization problem under a single-label, multi-class scenario. This problem could be defined as: given a training set of documents $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_c]$, organized in c categories, where $\mathbf{D}_i = [\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \ldots, \mathbf{d}_{i,m}]$ corresponds to a term-document (category) matrix composed by m documents in the i-th category; and given a test document $\mathbf{y}$. The goal is to train a model over the set $\mathbf{D}$ which is able to assign *one* of the c categories to the new test document $\mathbf{y}$.

We present in this work a novel document classifier which uses a framework that derives PCA from the minimization of reconstruction error of the training examples. The model uses such property to classify new documents, expecting than documents are better reconstructed by a transformation matrix which was computed using similar documents. We call this method the Minimum Loss of Reconstruction Information (mLRI) classifier. During training, the mLRI classifier computes a set of category transformation matrices $\mathbf{W}_i$;$i = 1, 2, \ldots, c$ of

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 91–103, 2013.

rank r by means of PCA. In the mLRI classifier, a suitable rank r is learned from the training data using a standard k-fold cross validation. During testing, given a new unseen document, the model projects such document using the different category matrices, then reconstructs the document using the same matrices and finally computes the reconstruction errors, by measuring the Frobenius norm of the difference between the set of reconstructed documents and the original one. The matrix which produces the minimum error indicates the category to be assigned to the new document.

We test the mLRI classifier with several public well-known datasets for text categorization: two subsets of the Reuters-21578 dataset, the Classic dataset and the 20Newsgroups dataset. We test the model using standard training/test split of the data. In addition we test the ability of the model to generalize the categories by systematically reducing the amount of training documents and keeping all the test document in the 20Newsgroups dataset. The results show that mLRI gives good and stable results across the different datasets and experiments. In order to have a better overview of the performance of the mLRI classifier, we present a comparison for every experiment with three other well known categorization methods: Multinomial Naive Bayes (NB), K-Nearest Neighbors (KNN) and the SMO classifier, a popular Support Vector Machine (SVM) with a very good behavior in text document categorization.

The contributions of our work are: the feasibility of applying the framework of minimizing the reconstruction error using PCA in a single-label multi-class text categorization task, exploiting the sparseness of the category matrices in order to perform a fast training of the model. The empirical evidence that mLRI is able to properly model the categories of the documents by extracting transformation matrices which represent most of the information from the data in terms of variance and minimization of the reconstruction error of the training documents. We evidence that the property of minimizing the reconstruction error could be extended and applied to unseen documents and that a suitable rank r for the projection matrices $\mathbf{W}_i$ could be learned from the training data.

The remainder of this paper is organized as follows: in Section 2 we first give a brief introduction to PCA, from which we derived the mLRI model, and secondly we describe the general architecture for mLRI. Section 3 is devoted to review several statistical and probabilistic works related to mLRI. In Section 4 we give the experimental evaluation framework for the mLRI classifier, describing the datasets and setup used during experimentation, and we present the results together with a discussion about them. Finally, in Section 5 we present the conclusions and future work.

2 Proposed Model

2.1 PCA

Principal Components Analysis (PCA) is one of the most popular methods for approximating a given dataset using a linear combination of the original features, producing a compressed representation of the data by using fewer combined features [1][13][16][23]. PCA was first developed by Pearson [23] and its statistical properties were investigated in detail by Hotelling [13].

There are several frameworks from which we can derive PCA [21][14]. The most popular one is by an analysis of variance. The derivation of PCA in this framework generates the basis by iteratively finding the orthogonal directions of maximum retained variances [13][16]. The other way of deriving PCA is using the minimization of the mean-square error in data compression [4][7][14]. In this approach, PCA finds a lower-dimensional linear representation of the data, such that the original data could be reconstructed from the compressed representation with the minimum square error. This second formulation is the one of interest in this work, and from which we derive the mLRI classifier. The brief formulation of PCA from data compression using documents as data is given below, for a larger explanation the reader is referred to [4] and [7].

Let $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m]$ be a term-document matrix of training data, were $\mathbf{d}_i \in \mathbb{R}^n$ is the i-th document expressed as a *column* vector with n rows. The rows represent the terms from a vocabulary.

PCA looks for a matrix $\mathbf{W} \in \mathbb{R}^{n\times r} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_r]$, composed by a set of $r \leq n$ orthogonal *column* vectors, with unit length, which transforms the original documents to a new orthogonal basis. We called such vectors *principal components* (PCs). The transformation is expressed as: $\mathbf{z}_i = \mathbf{W}^T(\mathbf{d}_i - \overrightarrow{\mu})$;$i = 1, 2, \ldots, m$, where $\mathbf{z}_i$ is the i-th projected document in the new space, and $\overrightarrow{\mu}$ is the mean vector of the term-document matrix. Inversely, we can express the original documents as: $\mathbf{d}_i = \mathbf{W}\mathbf{z}_i + \overrightarrow{\mu}$;$i = 1, 2, \ldots, m$

Under the data compression approach, PCA finds $\mathbf{W}$ such that the Frobenius norm of the reconstruction $f = \sum_{i=1}^{n} \|\mathbf{d}_i - (\mathbf{W}\mathbf{z}_i + \overrightarrow{\mu})\|_F$ is minimized.

If we choose $r = n$ as the dimension of $\mathbf{W}$ we do not lose any information during the transformations, and the error is zero. However, if we choose $r < n$, some information is lost because not all the variance is present. The choosing of a $r < n$ is common for *data compression* and dimensionality reduction, because the largest variances are associated with only the r first components. Then, the error of reconstruction for a given dimension $r < n$ is $e = \sum_{i=r+1}^{n} \mathbf{w}_i^T \mathbf{A}\mathbf{w}_i$, where $\mathbf{A} = \sum_{i=1}^{m} (d_i - \overrightarrow{\mu})(d_i - \overrightarrow{\mu})^T$ is the covariance matrix of the mean centered data matrix. The minimization of this error corresponds to the following eigenproblem: $\mathbf{AW} = \lambda\mathbf{W}$.The reconstruction error is then minimized when $\mathbf{w}_i$ are the eigenvectors of the covariance matrix. This will produce the optimal low rank r approximation to $\mathbf{D}$ in the least squares sense, with r as the rank of matrix $\mathbf{W}$, corresponding to non-zero eigenvalues.

2.2 Minimum Loss of Reconstruction Information

PCA minimizes the reconstruction error of the original documents, the idea behind the proposed approach is to extend such property to new unseen documents. Thus, if we have a matrix $\mathbf{W}_p$ that was obtained from a given category p of documents, this $\mathbf{W}_p$ must well reconstruct new unseen documents which are *similar* to the ones used to extract such matrix, and will reconstruct poorly unseen documents which are not similar or not close by related to the original ones. Using such assumptions we build the Minimum Loss of Reconstruction Information (mLRI) classifier, which generalizes the concept of error reconstruction to a single-label, multi-class scenario, where the task is to assign an unseen document to *one* of a predefined set of c categories, by finding the matrix $\mathbf{W}_p$ (and its corresponding category p) that better reconstructs the new document.

PCA is a crucial part of the system, then we consider some practical issues in order to make this computation more efficient. First, rather than compute the matrix $\mathbf{W}$ from the covariance matrix (which is expensive in both, memory and computing time), the mLRI classifier uses the relation of PCA with the Singular Value Decomposition (SVD), where the PCs correspond with the singular vectors of a SVD of the normalized centered data: $\texttt{SVD}((\mathbf{D} - \mathbf{M})/m^{\frac{1}{2}})$, which is more efficient [26]. Second, usually the rank r of the matrix $\mathbf{W}$ is chosen to be small in comparison with the total number of terms ($r \ll n$), since only a few components carry most of the information from the data. We use here the Power Factorization Method (PFM) [11][22], which is a generalization of the Power Method [13], as a SVD calculator to extract $\mathbf{W}$ with a relatively small rank r from the data matrix. The PFM starts from an initial random matrix $\mathbf{W}_{initial}$ of rank r and iteratively refines it until it reaches convergence using an orthonormalization of the columns [11]. The PFM converges very quickly to the desired approximation of $\mathbf{W}$ using simple matrix multiplications. Finally, for text document categorization, the term-document matrix $\mathbf{D}$ is highly sparse. Even if during the mean subtraction, part of the sparsity is lost, the use of a global vocabulary (i.e. extracted from all the categories in the training dataset), and the computing of PCA per category, leaves the data matrix still highly sparse. We exploit such sparseness in order to compute efficiently the low rank matrix $\mathbf{W}$.

The general architecture of the mLRI method is divided in a training and a test phase. The training phase starts with a set $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_c]$ of training documents organized in c categories. $\mathbf{D}_i = [\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \ldots, \mathbf{d}_{i,m}]$, corresponds to a term-document (category) matrix composed by m document column vectors of size n in the i-th category. The number m changes depending on i (different number of documents inside different categories). Then, we compute the mean vector $\overrightarrow{\mu}_i$ for each category i, and we form a matrix $\mathbf{M}_i \in \mathbb{R}^{n \times m}$ by repeating in each column the value of the mean $\overrightarrow{\mu}_i$. Afterwards, we subtract the corresponding mean to each category matrix $\mathbf{D}_i - \mathbf{M}_i$ and normalize the result by the square root of the number m of documents. Finally, we define the number r of PCs we want to extract from the data and compute the rank r matrix $\mathbf{W}_i$ for each category i using PCA from SVD. The parameter r defines the quantity of information carried by the matrix $\mathbf{W}$, but not all the information is important in such matrix. Then, the selection of a suitable value for r is important and depends on the properties of the data at hand. Such parameters could be learned in the mLRI classifier using standard techniques, like a k-fold cross validation over the training set. The training phase is illustrated in algorithm 1.

The test phase of the mLRI classifier is as follows: starting with a new unseen document $\mathbf{y} \in \mathbb{R}^{n \times 1}$ to be classified, expressed as a column vector, for each category i we first extract from the new document the category mean vector $\overrightarrow{\mu}_i$. We

Algorithm 1. Training phase for mLRI

```
Require:  – A set of training document matrices $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_c]$, organized in $c$ categories.
    – The size of the low rank approximation $r$.
Ensure:  – A set of projection matrices $\mathbf{W}_i$ ; $i = 1, 2, \ldots, c$.
    – A set of mean vectors $\overrightarrow{\mu}_1, \overrightarrow{\mu}_2, \ldots, \overrightarrow{\mu}_c$. One mean per category.
  for $i = 1$ to $c$ do
    $\overrightarrow{\mu}_i \leftarrow$ ComputeVectorMean($\mathbf{D}_i$)
    $\mathbf{M}_i \leftarrow$ RepeatMeanInColumns($\overrightarrow{\mu}_i$)
    $\mathbf{T}_i \leftarrow (\mathbf{D}_i - \mathbf{M}_i)/m^{\frac{1}{2}}$
    $\mathbf{W}_i \leftarrow$ SVD($\mathbf{T}_i, r$)
  end for
```

then project the document using the corresponding transformation matrix $\mathbf{W}_i$. Afterwards, we reconstruct back the document using the same matrix $\mathbf{W}_i$ and we add again the category mean vector $\overrightarrow{\mu}_i$. We then compute the reconstruction error as the Frobenius norm of the difference between the original document and the reconstructed one. The process is repeated for each category $i = 1, 2, \ldots, c$. Finally, the classifier assigns the document to the category with the minimum reconstruction error. This process is detailed in algorithm 2.

Algorithm 2. Testing phase for mLRI

```
Require:  – A set of projection matrices $\mathbf{W}_i$ ; $i = 1, 2, \ldots, c$.
    – A set of mean vectors $\overrightarrow{\mu}_1, \overrightarrow{\mu}_2, \ldots, \overrightarrow{\mu}_c$.
    – A new unseen document $\mathbf{y}$
Ensure:  – A predicted category $p \in \mathbf{C}$.
  for $i = 1$ to $c$ do
    $\mathbf{t}_i \leftarrow \mathbf{y} - \overrightarrow{\mu}_j$
    $\mathbf{z}_i \leftarrow \mathbf{t}_i \mathbf{W}_i$
    $\mathbf{s}_i \leftarrow \mathbf{z}_i \mathbf{W}_i^T$
    $\mathbf{q}_i \leftarrow \mathbf{s}_i + \overrightarrow{\mu}_i$
    $e_i \leftarrow$ FrobeniusNorm($\mathbf{y}$,$\mathbf{q}_i$)
  end for
  $p \leftarrow$ IndexofMinimum($e_1, e_2, \ldots, e_c$)
```

3 Related Works

On of most common techniques for dimensionality reduction in text mining is Latent Semantic Analysis (LSA) [6]. LSA uses the term-document matrix of a collection of documents and finds a low-rank approximation of such matrix using SVD. In matrix terms, LSA is similar to PCA, but without centering the term-document matrix. In line with this approach, most of the works devoted to text categorization where PCA or LSA are used, employ PCA as a first step to project the original term-document matrix to a new low-rank space, considering only the first few components with the highest variance. After this initial dimensionality reduction phase the categorization is performed using standard classification algorithms (e.g., SVM, NB, K-NN, etc.) [17][18][25]. Inside the mLRI classifier, we do not use the original training documents projected to a low-rank space to train a model, rather we use the transformation matrices computed from such original training documents as the classification model.

Linear Discriminant Analysis (LDA) is a categorization/dimensionality reduction technique [8], which uses the category information to project the data into a new space where the ratio of between-class-variance to within-class-variance is maximized in order to obtain adequate category separability [1]. Torkkola [27] was one of the first authors to use LDA for text categorization. There, the author mentions that PCA does not help for an optimal discrimination of the data, and then proposes LDA to categorize text documents. Nevertheless LDA as classifier tends to perform worse than a SVM for text categorization [17]. In this work, we actually exploit the discriminative properties of PCA for text categorization, by including the category information in the form of a transformation matrix per category, which minimizes the reconstruction error of the training documents of the corresponding category.

Non-Negative Matrix Factorization (NMF) is another dimensionality reduction technique, which projects the data to a new space, but the values in the transformation matrices obtained with NMF are only positive [2][3]. NMF is similar to other models such as probabilistic Latent Semantic Analysis (pLSA) [12] and Latent Dirichlet Allocation [5], which are probabilistic extensions of the LSA model. These models reduce the dimensions of the documents by representing them as a mixture of topic distributions and topics as a mixture of word distributions. These models have the disadvantage that identifying the correct number of latent components is a difficult and computationally expensive problem [5]. As we will show further, when using the mLRI classifier, we could estimate the proper number of latent components (the rank r of the matrix W), by applying a k-fold cross validation over the training set.

In this work, we proposed the mLRI classifier, which relies on the framework derived from the property of PCA to minimize the reconstruction error of the training documents used to compute the matrix $\mathbf{W}$ and extend such property to unseen documents. The same property of PCA has already been used in computer vision tasks such as object detection [19] and novelty detection [12], and in a spam filtering task [9]. However, to the best of our knowledge there is no work devoted to minimize the reconstruction error on single-label multi-class text categorization, where the best rank r for the matrix $\mathbf{W}$ is learned from the training data.

4 Experimental Evaluation

4.1 Datasets

The following sections contain brief descriptions of the datasets used to test the validity of the mLRI classifier.

- **R8 and R51**: These two datasets are subsets of the well-known Reuters-21578 corpus using the “modApté” train/test split[1]. We removed from the

[1] Available at: `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

R10 (the set of the 10 categories with the highest number of positive training examples) and the R90 (the set of the 90 categories with at least one positive training example and one positive test example) subsets all the examples with zero or more than one category assigned, plus the categories with less than 2 examples for training (this is done with the purpose of applying the mLRI classifier, which captures information from a set of documents). This cut leaves 8 out of the 10 original categories in R10, and 51 out of the 90 in R90. The names are thus R8 and R51.

- **Classic**: This dataset[2] is a collection of abstracts from journals in different areas split in 4 categories.
- **20Newsgroups**: This dataset is a collection of approximately 20,000 newsgroup documents, split across 20 different newsgroups. In this work we used the *bydate* version of this dataset[3], which already has a train/test split.

We applied a pre-selection of terms from the training set in order to form a vocabulary for each dataset. We removed first the stop-words using the Rainbow list [20] and later we removed the terms that appear in only one document in the training set. This pre-selection helps to remove noisy features but avoiding an aggressive selection of features, which could remove some important features. Moreover, it is known that for classifiers that exhibit a good performance in text categorization, like a SVM or NB, the use of a large vocabulary is beneficial to be able to map between features and categories [15]. After the removal of terms, some documents from the Classic and 20Newsgroup dataset were empty and then removed from the dataset. Table 1 presents a summary of the datasets, which includes the split of training/test documents used, the number of categories and the total number of individual terms in each dataset.

Table 1. Summary of the datasets used for experimentation

Dataset	Number of Training Documents	Number of Test Documents	Number of Categories	Vocabulary Size
R8	5485	2189	8	19287
R51	6531	2566	51	21566
Classic	4956	2123	4	20177
20NewsGroups	11293	7528	20	72579

4.2 General Experimentation Setup

The term-category matrices were built by vectorizing the text documents using a term-frequency inverse-document-frequency (tf-idf) schema. The tf-idf vectors are normalized to the unit using norm 2. The documents are in the columns and the terms in the rows. All the methods start the training process using directly the normalized tf-idf term-document matrices.

[2] Available at: `http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/`

[3] Available at: `http://people.csail.mit.edu/jrennie/20Newsgroups/`

In order to have a better overview of the performance of the mLRI classifier, we compared its results with the ones obtained using other popular classification algorithms. We used for comparison three classifiers: Multinomial Naive Bayes (NB), K-Nearest Neighbors (K-NN), and SMO [24], a popular linear Support Vector Machine (SVM) which is known by its very good performance in text categorization. We used for these baseline methods the implementations from the Weka package [10] using Java to call the methods programmatically. In order to find the optimal parameters C for SVM (C is the penalty constant, a larger C corresponding to assigning a higher penalty to errors) and K for K-NN (K is the number of nearest neighbors considered to assign the category to a test document), we performed a 5-fold cross validation over the training set with different parameter values: $C = \{0.01, 0.1, 1, 10, 100, 1000\}$ and $K = \{2, 5, 10, 20, 30, 50\}$ and we choose the one which maximizes the macro-F1 measure.

The model for mLRI was implemented in MatLab. The PFM method approximates the matrices $\mathbf{W}_i$ starting from a random matrix; thus, in order to estimate the confidence level of the approximations, we performed 10 runs with mLRI for each experiment. Similarly than with the SVM and the K-NN classifiers, with the mLRI classifier we performed a 5-fold cross validation over the training data, in order to optimize the r parameter, which corresponds to the rank of the matrices $\mathbf{W}_i$. We considered ranks of $r = \{1, 2, 4, 8, 16, 32, 64, 128\}$, and we chose the one which maximizes the macro-F1 measure. Additionally, we show the results of the effect of changing the rank r of the matrices $\mathbf{W}_i$ in the reconstruction error over the test set.

We compared the performance of all the methods using the standard classification measures accuracy and F1-measure, which are defined as $Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$ and $F1 = 2\frac{Precision \cdot Recall}{Precision+Recall}$. Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. The accuracy measures the proportion of corrected categorized documents, while the F1 measure represents the harmonic mean of precision and recall. We compute the micro- and macro-averages for F1. In our case, given that the categorization is single-label multi-class, the values of F1 micro-averaged are equal to the values of accuracy. We conducted all the experiments using a desktop Linux PC with a 3.4 Ghz Intel Core i7 processor and with 16 GB in RAM.

4.3 Results

Tables 2, 3, 4 and 5 show the results of all the classifiers for the R8, R51, Classic and 20Newsgroups datasets respectively. The tables present in the first column the name of the method used. In the second column appears the number of features. In the case of the mLRI classifier this number corresponds to the rank r of the matrix $\mathbf{W}$ used to perform the error reconstruction; while in the case of the other methods this number corresponds to the size of the vocabulary. In column three to six of the tables appear the accuracy, the macro-F1, the training time and the testing time, respectively. The accuracies and macro-F1 measures for the mLRI classifier are averages over the 10 runs performed in each

Table 2. Comparison of performance for the R8 dataset

	Number of		**Macro-**	**Training**	**Testing**
Method	**Features**	**Accuracy**	**F1**	**Time (s)**	**Time (s)**
mLRI	1	*0.964±1E-3*	*0.922±2E-3*	2.55	2.9
	2	0.951±1E-3	0.920±2E-3	2.65	3.14
	4	0.952±1E-3	0.918±1E-3	2.97	3.43
	8	**0.949±2E-3**	**0.917±2E-3**	3.98	4.16
	16	0.952±1E-3	0.918±2E-3	5.84	5.56
	32	0.949±1E-3	0.909±2E-3	9.86	8.41
	64	0.940±2E-3	0.908±2E-3	19.07	13.92
	128	0.928±1E-3	0.904±7E-4	42.70	24.57
SVM (C=0.01)	19287	0.940	0.895	2.06	1.16
NB	19287	0.804	0.351	0.2	0.08
KNN (K=2)	19287	0.595	0.402	-	14.715

Table 3. Comparison of performance for the R51 dataset

	Number of		**Macro-**	**Training**	**Testing**
Method	**Features**	**Accuracy**	**F1**	**Time (s)**	**Time (s)**
mLRI	1	*0.926±2E-3*	*0.693±2E-3*	2.94	5.26
	2	0.919±1E-3	0.692±4E-3	3.04	5.65
	4	0.919±1E-3	0.689±3E-3	3.49	6.27
	8	**0.918±2E-3**	**0.691±2E-3**	4.83	7.74
	16	0.917±2E-3	0.684±3E-3	7.45	10.27
	32	0.913±2E-3	0.668±1E-3	13.38	15.13
	64	0.901±1E-3	0.644±4E-3	28.25	23.17
	128	0.876±1E-3	0.594±6R-4	71.78	36.83
SVM (C=0.1)	21566	0.903	0.702	18.38	36.13
NB	21566	0.684	0.053	0.2	0.1
KNN (K=2)	21566	0.549	0.335	-	21.97

Table 4. Comparison of performance for the Classic dataset

	Number of		**Macro-**	**Training**	**Testing**
Method	**Features**	**Accuracy**	**F1**	**Time (s)**	**Time (s)**
mLRI	1	0.947±1E-3	0.950±1E-03	1.77	2.64
	2	0.950±2E-3	0.954±2E-3	1.85	2.85
	4	0.952±2E-3	0.956±2E-3	2.13	3.14
	8	0.956±2E-3	0.960±2E-3	2.93	3.74
	16	**0.957±1E-3**	**0.962±1E-3**	4.41	4.92
	32	*0.958±1E-3*	*0.962±1E-3*	7.44	7.35
	64	0.958±6E-4	0.961±6E-4	13.98	12.18
	128	0.957±6E-4	0.959±5E-4	28.95	22.02
SVM (C=0.1)	20177	0.954	0.956	2.48	0.45
NB	20177	0.938	0.938	0.2	0.2
KNN (K=2)	20177	0.458	0.171	-	14.60

experiment. These measures include the 95% confidence intervals with 9 degrees of freedom, estimated from a t-distribution. The training and testing times for the mLRI classifier are also averages over the 10 runs. The rows of the methods SVM and K-NN show the optimal parameters (C or K) found during the 5-fold cross validation and used with the test set. The training times for the K-NN method are always zero since it is a *lazy* classifier. The accuracies and macro-F1 measures in bold for the mLRI classifier correspond to the values obtained with the use of the optimized r, found with the 5-fold cross validation over the training set. The values in *italic* correspond to the best values in general.

Table 5. Comparison of performance for the 20 Newsgroups dataset

Method	Number of Features	Accuracy	Macro-F1	Training Time (s)	Testing Time (s)
mLRI	1	0.775±1E-3	0.770±1E-3	9.86	64.21
	2	0.777±2E-3	0.771±2E-3	10.10	68.75
	4	0.780±1E-3	0.774±1E-3	11.02	75.38
	8	0.787±1E-3	0.781±1E-3	14.08	89.35
	16	0.798±1E-3	0.792±1E-3	19.86	116.83
	32	0.811±1E-3	0.804±1E-3	33.53	172.32
	64	0.821±1E-3	0.814±1E-3	70.59	293.45
	128	**0.833±1E-3**	**0.827±1E-3**	177.75	536.66
SVM (C=1)	72579	0.803	0.800	98.45	68.09
NB	72579	0.796	0.774	0.2	0.2
KNN (K=2)	72579	0.114	0.112	-	261.19

R8 and R51 datasets: from table 2 we observe that mLRI performs better, in terms of both accuracy and macro-F1, than the other methods for the R8 dataset using a matrix of rank 8. The optimized rank r found by the 5-fold cross validation produces slightly worse results than the best rank in general (rank 1), but the performance is still appropriate. The training and testing times using a matrix of rank 8 are usually greater than the ones of NB and the SVM, but are relatively low. In table 3 we observe that for the R51 dataset, similarly than with the R8, the mLRI method performs good with a matrix of rank 8, surpassing the other methods in accuracy, but with a macro-F1 less than the one of the SVM. We observe the same effect of choosing an optimal rank r from the 5-fold cross validation, the results are not the best, but appropriate. Again, the training and testing times are greater than the ones of NB but in this case better than the ones of the SVM.

Classic dataset: in table 4 we observe that the results for the mLRI classifier with a matrix of the optimal rank found from the training set (rank 16) are better than all the other methods in accuracy and macro-F1. The optimal rank found is very close to the best general performance (rank 32). In this case, the training and testing times are higher than the ones of NB and the SVM, but are still low. In this case, the use of a matrix with rank $r = 16$ to perform the reconstruction of similar documents, means that more PCs are required to better express the variance inside the Classic dataset. This is perhaps due to more diversity inside each category and more overlapping content among categories.

20Newsgroups dataset: in table 5 we can see that the mLRI classifier performs better than the other methods, in terms of both accuracy and macro-F1, when using a matrix of rank 128. In this case, the optimal rank corresponds with the best general performance. The training and testing times are higher than the ones of the other methods, but still feasible. 20Newsgroup is a denser and balanced dataset than the other ones, but contains a greater overlapping of content inside them. That is one of the reasons this dataset is harder to categorize than the previous datasets, but still, mLRI models better the different categories than the other methods.

We performed a final set of experiments in order to test the ability of the methods to generalize the content of the different categories. We do this by

Table 6. Comparison of performance per training size in the 20 Newsgroups dataset

Training Size (%)	mLRI		SVM		NB		KNN	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	**0.472±1E-03**	**0.448±1E-03**	0.316	0.322	0.309	0.324	0.051	0.005
5	**0.696±4E-04**	**0.687±5E-04**	0.682	0.680	0.354	0.659	0.624	0.048
10	**0.741±9E-4**	**0.733±9E-4**	0.709	0.714	0.709	0.678	0.064	0.034
20	**0.779±8E-04**	**0.771±8E-04**	0.756	0.754	0.748	0.721	0.085	0.067
30	**0.797±7E-04**	**0.790±7E-04**	0.770	0.770	0.758	0.733	0.099	0.094
40	**0.803±4E-04**	**0.797±4E-04**	0.780	0.777	0.769	0.745	0.103	0.105
50	**0.822±6E-4**	**0.815±5E-4**	0.787	0.785	0.782	0.759	0.116	0.123
100	**0.833±1E-3**	**0.827±1E-3**	0.803	0.800	0.796	0.774	0.114	0.112

selecting fractions of the training set from the 20Newsgroups dataset, and we use the complete test set for testing. The results for these experiments are summarized in table 6. The first column of table 6 shows the proportion of the training set used, and the rest of the columns show the performance for accuracy and macro-F1 for all the methods. The results for the mLRI, SVM and K-NN correspond to the ones of the optimal r, C and K parameters, respectively, obtained with 5-fold cross validation over the training set. From this table we observe that the performance of the mLRI classifier is maintained disregarding the size of the training set used. This shows the ability of the method to generalize and properly model the categories inside the dataset.

In all the experiments, the estimated confidence levels are very low, meaning that the PFM is able to compute the low rank matrices $\mathbf{W}_i$ very consistently.

We observe from the experiments that the selection of the rank r for the transformation matrices $\mathbf{W}_i$ is an important issue depending on the dataset at hand. Similar to other parameters like C for SVM or K for K-NN, the r parameter needs to be tuned or adjusted to reach a good classification performance. We have seen from the experiments that this could be done by performing a standard k-fold cross validation over the training set.

5 Conclusion

In this paper we have presented and evaluated a novel technique to classify documents, using their text content features, in a single-label multi-class scenario. The proposed model relies on the property of PCA to minimize the reconstruction error of the documents used to compute a low-rank transformation matrix $\mathbf{W}$. We called the proposed method Minimum Loss of Reconstruction Information (mLRI) classifier. The mLRI classifier extends the property of error minimization of PCA and applies it to new unseen documents. We have shown that this technique is able to well preserve the diversity of the data from the different categories inside the transformation matrices $\mathbf{W}_i$. Then when such matrices are used to classify a new unseen document, the category matrix with similar properties to the new document, enables to reconstruct such document with minor loss of information. The reconstructed document based on a given category matrix that is closest to the original document indicates the category of the new document.

Results have shown that the mLRI classifier performs well in terms of accuracy (or micro-F1) and macro-F1, and when the rank of matrix $\mathbf{W}$ is small, its training and testing times are competitive to the ones of a linear SVM. During the training phase, mLRI takes advantage from the sparsity of term-document matrices and the good performance of the PFM, in order to fast approximate the low rank category matrices $\mathbf{W}_i$. In this sense, the selection of the rank parameter r is an important issue for the mLRI classifier, since depending on the dataset a different number of PCs are required to properly model the data in terms of minimization of the reconstruction error. Such parameter r is learned from the training documents by using a standard k-fold cross validation.

In the future we want to apply the mLRI method to other text classification tasks. In particular we are interested in using it for hierarchical classification in large datasets, where there exist thousands of documents and thousands of categories and the categories are arranged in a taxonomy. The mLRI could be used inside the taxonomy to model the internal categories. It would be interesting as well, to combine the output prediction of the mLRI with other classifiers.

Acknowledgments. This research was supported partially by the CONACYT postdoctoral grant I0010-2010-01/150813 and by the Belgian national project Radical (GOA 12/003).

References

1. Anderson, T.W.: An Introduction to Multivariate Statistical Analysis, 3rd edn. Wiley, New Jersey (2003)
2. Barman, P.C., Iqbal, N., Lee, S.-Y.: Non-negative Matrix Factorization Based Text Mining: Feature Extraction and Classification. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4233, pp. 703–712. Springer, Heidelberg (2006)
3. Berry, M.W., Gillis, N., Glineaur, F.: Document classification using nonnegative matrix factorization and underapproximation. In: Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 2009, pp. 2782–2785. IEEE (2009)
4. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (August 2006)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience (November 2001)
8. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
9. Gomez, J.C., Moens, M.F.: PCA document reconstruction for email classification. Computational Statistics & Data Analysis 56(3), 741–751 (2012)

10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
11. Hartley, R.H., Schaffalitzky, F.: PowerFactorization: 3d reconstruction with missing or uncertain data. In: Proceedings of the Australia-Japan Advanced Workshop on Computer Vision, AJAW 2003 (2004)
12. Hoffmann, H.: Kernel PCA for novelty detection. Pattern Recognition 40(3), 863–874 (2007)
13. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24(7), 498–520 (1933)
14. Ilin, A., Raiko, T.: Practical approaches to principal component analysis in the presence of missing values. Journal of Machine Learning Research 99, 1957–2000 (2010)
15. Joachims, T.: Text Categorization with Suport Vector Machines: Learning with many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
16. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer (October 2002)
17. Kim, H., Howland, P., Park, H.: Dimension reduction in text classification with support vector machines. Journal of Machine Learning Research 6, 37–53 (2005)
18. Li, Y.H., Jain, A.K.: Classification of text documents. The Computer Journal 41(8), 537–546 (1998)
19. Malagón-Borja, L., Fuentes, O.: Object detection using image reconstruction with PCA. Image Vision Computing 27(1-2), 2–9 (2009)
20. Mccallum, A.K.: BOW: A toolkit for statistical language modeling, text retrieval, classification and clustering (1996), `http://www.cs.cmu.edu/~mccallum/bow/`
21. Miranda, A.A., Borgne, Y.A., Bontempi, G.: New routes from minimal approximation error to principal components. Neural Processing Letters 27(3), 197–207 (2008)
22. Morita, T., Kanade, T.: A sequential factorization method for recovering shape and motion from image streams. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(8), 858–867 (1997)
23. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2(6), 559–572 (1901)
24. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods, pp. 185–208. MIT Press (1999)
25. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
26. Shlens, J.: A tutorial on principal component analysis. In: Systems Neurobiology Laboratory, Salk Institute for Biological Studies (2005)
27. Torkkola, K.: Linear discriminant analysis in document classification. In: Proceedings of the IEEE International Conference on Data Mining ICDM 2001 Workshop on Text Mining. IEEE (2001)

Semantic Classification of Posts in Social Networks by Means of Concept Hierarchies

Karina Ruiz-Mireles, Ivan Lopez-Arevalo, and Victor Sosa-Sosa

CINVESTAV-Tamaulipas, Victoria, Mexico
{aruiz,ilopez,vjsosa}@tamps.cinvestav.mx

Abstract. Social networks are in constant growth, here users share all kind of information such as news, pictures and their personal opinions about different topics. In order to a user can retrieve such content for a topic of interest, it must provide the terms believed to occur in the posts; but in a matter of semantics, this tends to leave out relevant results. This paper proposes an approach to perform semantic classification of posts in social networks using concept hierarchies (CH). This classification is considered as a first step towards semantic searching. In addition, a method to obtain a CH for a particular subject is also proposed. With the implementation of this approach, the obtained results reflect what it seems to be a so promising approach, obtaining more than 64% of accuracy on the F-measure.

Keywords: concept hierarchy, hierarchical classification, social network search, semantic classification, DBpedia, Linked Data.

1 Introduction

With the increasing amount of users in social networks, the published information is constantly growing. Usually, users search for information like news or opinions of other users, but at present, this search is made in a syntactic manner. So, the user must provide the terms he believes that could get accurate results based on the occurrence of words in the posts. This makes possible that if the user does not provide a representative word, related to the subject of interest, some relevant posts would be excluded from the results. This is an issue that can be well addressed by semantic search, which considers the meaning of words in a specific context (semantic of words).

In this paper we propose an approach to perform semantic classification of posts in social networks, considering as a case of study *Twitter* because it counts with a large amount of users and posts. The proposed approach is based on the use of a concept hierarchy for a specific domain.

A concept hierarchy (CH) is a conceptual graph, where its concepts (nodes) are organized in a hierarchical manner. This is accomplished by relations which determine the membership or generalization of concepts. A CH is a knowledge representation for a specific domain of interest.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 104–114, 2013.

In the presented approach, a text classification is applied to each post, assigning one or more concepts of the CH. This classification is made trough a voting system. To classify a post, it must be in English and related to the domain of the CH. It is also introduced a semiautomatic method to obtain the CH from a specific domain, by making use of the Semantic Web, specifically DBpedia, that follows the Linked Data principles, and it is one of the largest dataset of cross-domain in the Semantic Web.

The paper is organized as follows: Section 2 reviews related works, Section 3 presents the proposed approach. Implementation and experimental results are described in Section 4. Finally, Section 5 gives a conclusion.

2 Related Work

The information on Linked Data has been used in works that required semantic analysis. Waitelonis and Sack [1] proposed a method to search videos, obtaining data related to the query from Linked Data and managing it to improve the actual search engines. Hausenblas [2] described some applications based on Linked Data, one of them is Faviki[1], which allows users to assign tags to web pages by using DBpedia. Another application is the BBC music site[2], that obtains meta-data from Musicbrainz and artist information from DBpedia to provide suggestions to the user.

Most of the work about searching in social networks is performed about the users and not considering the information they share [3,4]. This might not be the best approach in the sense that a user can publish information about different topics. Reis *et al.* [5] proposed an approach for personalized semantic searching, using ontologies to represent shared meanings. They generated this ontology from the content of the social network, describing the social aspects shared by the users.

Regarding to the creation of the CH, there are some works focused on the automated generation of CH [6,7,8]. Some of them are based on a previous structure, while others perform a text analysis on a set of documents belonging to a particular domain. Yeh and Sie [6] combined an approach to automatically obtain a CH with a clustering algorithm based on a clustering procedure of votes. Sun and Lim [9] proposed a hierarchical classification based on a top-down approach. They build a binary classifier for each category (local classifier) and a binary classifier for each internal category (sub-tree classifier) to determine whether a document should be passed to its subcategories. Gelbukh *et al.* [10] presented an indexing representation method to map a document to a CH, assigning to each concept of the hierarchy a value that represents its relevance in the document. The assignment of the values to the concepts was made in two stages, first they assign a leaf node to the document and, in the second, they generalize this membership to the upper-level nodes.

[1] www.faviki.com

[2] www.bbc.co.uk/music

3 Approach

In this section, a methodology to obtain a CH based on DBpedia is presented. Also the obtaining of posts to analyze and assigning it a category from the CH is described. In order to present the approaches, the definition of a CH is given.

A basic conceptual hierarchy is a graph formed by a set of concept nodes, representing entities in the domain, and a set of relations representing the relationships between the entities[11]. The ordered set of concepts is denoted by T_C and the set of relations by T_R.

A concept hierarchy can be represented as $CH(T_C, T_R, T_T)$ where:

- T_C is the set of concept types. It is partially ordered by a relation $\leq$, having a greatest element denoted by $\top$.
- T_R is the set of relation symbols.
- T_T is the set of term types, each term belongs to n concepts.

3.1 Obtaining the Concept Hierarchy

DBpedia is a dataset containing information extracted from Wikipedia, following the Linked Data principles, which provides a classification schema for things. To create the CH we make use of the Wikipedia Categories, which are represented using the SKOS vocabulary[3] and DCMI terms[4]. Taking into account that for the hierarchy of concepts, terms and hierarchical relationships are required, the following is the data obtained from DBpedia:

- **Concept**. The Wikipedia categories.
- **Hierarchical Relation**. Which are obtained by using the `skos:broader` property.
- **Terms**. The things that are related to a concept by the property `dct:subject`.

We represent this information by using RDF triplets. Accessing the information in DBpedia, a concept is saved as follows:

```
<conceptURI> skos:broader <parentURI>
<termURI> dct:subject <conceptURI>
```

In order to be able to use this information and to allow the user to identify the concepts of the CH, we obtain the labels of the concepts and terms.

In DBpedia the categories of Wikipedia have a URI with the structure `http://dbpedia.org/resource/Category:Topic`, where *Topic* is the identifier of the concept. To extract the CH from DBpedia, we select a category as the root concept ($\top$) and domain of the CH. This is the starting point of a Breadth First Search exploration over the DBpedia graph. To obtain the concepts of the CH we analyze a concept obtaining the terms and the labels

[3] www.w3.org/2004/02/skos/

[4] www.dublincore.org/documents/dcmi-terms/

related to it (concept and its terms). Nevertheless, there is a problem with this approach, due to the large amount of nodes and relations in DBpedia, we can obtain concepts that are out of the desired domain. In order to solve this issue, the Normalized Google Distance (NGD) measure proposed by Cilibrasi and Vitanyi [12] was implemented:

$$NGD(x, y) = \frac{max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - min\{\log f(x), \log f(y)\}} \tag{1}$$

where:

- $f(x)$ is the number of pages containing x.
- $f(x, y)$ is the number of pages containing x and y.
- N is a normalization factor.

According to the experimentation, for the application of such measure is better to begin from level 3 of the DBpedia graph to evaluate the similarity of nodes with its ancestors. That is, we evaluate the similarity between the concept and all their grandparents with the NGD to obtain an average of this similarity. If this average is less than 0.9 (in NGD 0 indicates that the concepts are identical and 1 unrelated), the concept is added to the CH.

3.2 Obtaining Posts

In order to obtain posts we request the public stream of Twitter through the Java library Twitter4J[5], and filter it by a set of users. To obtain these users, we send HTTP requests to the WeFollow[6] site, setting as a parameter the desired topic; these requests are made iterating the results. Before adding a user to user set, we verified that it is a valid user account, if the user has their posts as public and if it has the account language set as English. This is because our method only works for posts in English and from a specific domain.

Once obtained the set of users, we begin to capture posts, but due to a user may have configured their account in English even when he writes its posts in another language, we filter the posts by selecting only those in English. In order to obtain the language of the text of a post, we use the API *Free Language Detection*[7].

Before classifying the posts, it is necessary a preprocessing of text. The language used in social networks includes irrelevant terms to the language, or even characters to express emotions in posts. The actions taken as part of the preprocessing of the posts are the following:

[5] twitter4j.org

[6] wefollow.com

[7] detectlanguage.com/

1. **Cleanup reference to other users**
   Links to other users are removed.
2. **Cleaning of hashtags**
   The hashtags are replaced by its meaning, which is obtained trough *tagdef*'s[8] API. By sending an HTTP request, a JSON answer is obtained with all the meanings that exists in *tagdef*.
3. **Conversion of abbreviations**
   A list of abbreviation-meaning is obtained from *webopedia*[9]. Every time an abbreviation from the list appears in a post, this is replaced with its meaning.
4. **Extraction of data from links**
   When a post contains a link to an external site, information about the web page is obtained. We extract the title, metadata keywords and description to add them to the text of the post.
5. **Text cleaning**
   Eliminate punctuation marks and stopwords, such as pronouns and prepositions.

3.3 Selection of Concepts

The assignment of concept(s) to a post is based on a voting system that consists on the following:

1. Search every word of the post in the terms contained in the concepts of the CH. Since the terms of the concepts in the CH can consist of one or more words, we state that a post contains a term if all its words appear in the post.
2. For each term (in the CH) that the post contains, a vote is assigned to all of the concepts to which the terms belongs to.
3. Additionally to this selection of terms, when 90% of the terms of a concept contain a word in the post, this concept is also selected.
4. Besides the votes assigned to the concepts depending on the terms, a vote is assigned to the concepts which label appears in the post.
5. After the votes are assigned, in order to determinate which concepts represent the post, the concept with more votes is assigned. In case of a draw between two or tree concepts, all of them are selected. If as a result of the voting system more concepts are selected, for each of these concepts we assign as many votes the concept has to their broader concept(s). This generalization of concepts is applied until tree or less concepts obtain the maximum number of votes.

In the points 1 and 2 it is indicated how the concepts are selected, the points 3 and 4 are to select the possible concepts, and in the point 5 we indicate the selection of the concepts as from the votes assigned.

[8] `tagdef.com/`

[9] `www.webopedia.com/quick_ref/textmessageabbreviations.asp`

4 Preliminary Results

In this section the preliminary results of the applied approaches and its evaluation measures are presented.

4.1 Evaluation Measures

Since this approach is based on a hierarchical classification, in order to evaluate the obtained results, we make use of the hierarchical evaluation measures proposed by Kiritchenko *et al.* [13]: hierarchical precision (hP), hierarchical recall (hR) and hierarchical f-measure (hF), which are a generalization of the standard precision, recall and f-measure.

$$hP = \frac{\sum_i \mid \hat{C}_i \cap \hat{C}'_i \mid}{\sum_i \mid \hat{C}'_i \mid} \tag{2}$$

$$hR = \frac{\sum_i \mid \hat{C}_i \cap \hat{C}'_i \mid}{\sum_i \mid \hat{C}_i \mid} \tag{3}$$

where $\hat{C}'_i$ is a set consisting of the most specific class(es) predicted for the test example i and all its(their) ancestor concepts and $\hat{C}_i$ is the set consisting of the true most specific class(es) of the test example i and all its(their) ancestor concepts. Thus, the hF measure is defined as:

$$hF = \frac{2 * hP * hR}{hP + hR} \tag{4}$$

4.2 Running Examples

In order to test the proposed approach, we obtained a CH for the topic *Computers*. In this section we illustrate how the assignment of concepts works according to the proposed approach based on 3 scenarios.

Case 1
Starting from the post:

> Kindle Fire update finally brings password-protection for purchases and other parental controls http://t.co/XOd6p1LL #technology

In the preprocessing of the text, we obtain:

- The meaning of the hashtag *#technology* (tweets related to new technologies)
- The metadata from the link (title, keywords and description)

which are added to the original post, then punctuation marks and stopwords are eliminated. The resulting text for the original post is:

kindle fire update finally brings password protection purchases other parental controls technology tweets related new technologies kindle fire adds password protection purchases crave cnet software update version 6 3 1 parental controls amazon gives parents control over kids see purchase read blog post david carnoy

We analyze every word from the resulting text to determinate which terms from the CH appear in the post. In this case

- *update*
- *CNet Technology*
- *technology*
- *Parental controls*
- *purchase*
- *finally*
- *Blog*
- *Blog software*
- *Password*
- *version*
- software
- *read*
- *Amazon Kindle*
- *protection*
- *control*
- *amazon*

are some of the terms assigned to this post.

Each term votes for the concepts they belong to: for example, the term *Parental controls* assigns a vote to the concept *Internet safety,* and the term *Password* to the concepts *Password authentication* and *Privacy software.*

Besides the votes assigned to the concepts by their terms, some of the votes assigned to possible concepts for the original post are:

- *Blogging*
- *Computer hardware companies*
- *Password authentication*
- *Computer access control*
- *Concurrency control*

Some of the first voting results are the next:

Concept	**Votes**
Password authentication	4
Internet safety	4
Capcom	3
Smartbooks	3
...	

The selection of the concepts in this case is very simple, we only select the concepts with the most votes, *Password authentication* and *Internet safety.*

Case 2

Consider the following post:

Future iPhone May Let You Gift Tunes by Tapping Devices [VIDEO] - http://t.co/2bh54ExQ

The preprocessed text is:

future iphone may let gift tunes tapping devices video after applying the same voting system, we obtain the following five concepts

- *Social Media*
- *Blogs by Country*
- *Social networking Services*
- *Social Networks*
- *Spamming*

As established by the proposed method, we only assign a maximum of tree concepts per tweet. In order to reduce the number of selected concepts, we generalize this results by adding the votes of each selected concept to their parent(s). According to the five top selected concepts (nodes in gray) and following the relations of the CH shown in the Figure 1, the concepts *blogs by country*, *World Wide Web* and *Internet Culture* get one extra vote, while the concept *Social information processing* gets three, assigning it as the best concept for the post.

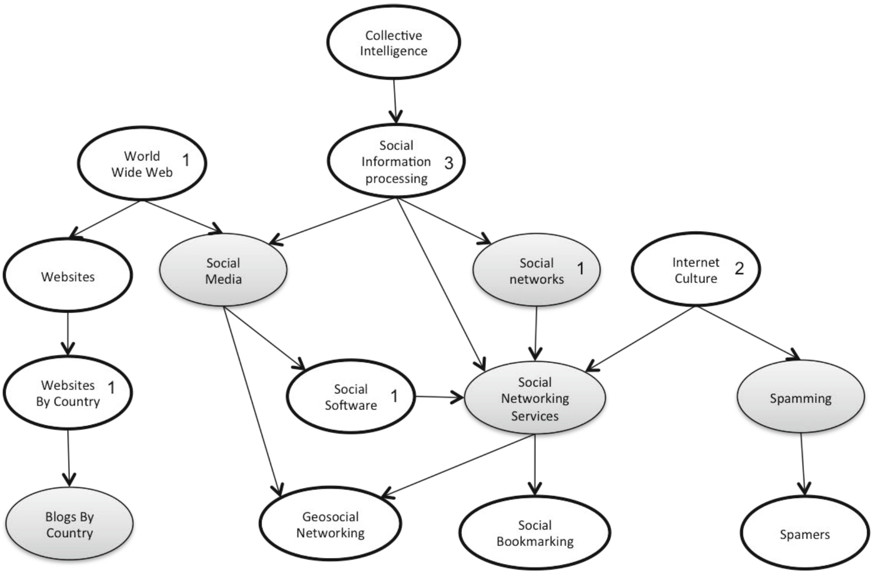


Fig. 1. Partial view of the CH for *Computers.*

Case 3

Sometimes the stream from Twitter contains non-related posts for a specified topic. In the case of a post non-related to the domain of *Computers* like the next:

> Mexican experts find ancient blood on stone knives http://t.co/Ht9EwnR3

The preprocessed text is:

mexican experts find ancient blood stone knives mexican experts find ancient blood stone knives

The method tries to map this text to the CH even though it is not its domain. As a result of the voting procedure we obtain, among others, the following concepts, with 2 votes each one:

- *Free panorama software*
- *Acid tests*
- *Domain names*
- *Acid tests*
- *Cancelled Linux games*
- *Mobile hotspots*
- *Acid tests*
- *Linux drivers*
- *Internet begging*
- *.NET game engines*
- *Parodies of Wikipedia*
- *Free industrial software*

As it can be seen, none of those concepts correspond to the post, so when we have this case that over 20 concepts tie. In this case, no concept is assigned to the post.

Evaluation

For the experimentation it was necessary to define a set of posts with the right concepts (according to the CH for *Computers*). This corpus consists of 3640 posts manually classified by a group of users from Twitter. The results of the proposed method were compared against this corpus showing acceptable values. The obtained results are shown in Table 1. It is important to remark that do not exists similar approaches for classification of posts in the literature.

Table 1. Results

Number of posts	hP	hR	hF
3,640	67.46%	61.25%	64.21%

The results reported by Kiritchenko *et. al.* [13] for hF (with three different datasets) are shown in the Table 2, even though their approach is different in the way that they propose a hierarchical learning algorithm.

Table 2. Kiritchenko results

Dataset size	hF
3,305	59.31 %
2,468	38.17 %
2,284	73.35 %

5 Conclusions

In this paper we presented an approach to classify posts from Twitter. The approach is an initial effort to develop a semantic search module for social networks.

This proposal is based on concept hierarchies extracted from DBpedia. At thie moment, the approach only works for posts in English and related to the subject of the concept hierarchy. The concept hierarchy is a graph where nodes represent the concepts and each concept has a series of terms related to it. According to the terms of the post, the concept hierarchy is traversed adding votes to concepts. The concept with the maximum of votes is selected as the right concept for the post. A maximum of 3 concepts was established for each post. Three scenarios for classification were tested: (1) when the concepts for a posts are clearly the most voted, (2) when a generalization process is performed for a post with several (more than 3) equal-voted concepts and (3) when the post is not related to the subject of the concept hierarchy.

We also presented an approach to obtain a concept hierarchy from DBpedia, confronting the problem of the inclusion of concepts out of the domain of interest by using a similarity distance based in Google results. This approach is semi-automated because it is necessary to indicate when to stop the retrieval of concepts; after that, it may be necessary to delete some concepts, which are not related to the desired topic.

According to the results in our experiments, we can observe that the implementation of the approach has a hierarchical F-measure value of 64.21%, which is a good result for our hierarchical classification considering that it improves two of the reported results from the original work where the used measures were proposed. Another of our contributions is a dataset of tweets manually tagged with concepts within the area of Computers, which continues growing.

References

1. Waitelonis, J., Sack, H.: Towards exploratory video search using linked data. In: 11th IEEE International Symposium on Multimedia, ISM 2009, pp. 540–545 (2009)
2. Hausenblas, M.: Exploiting linked data to build web applications. IEEE Internet Computing 13, 68–73 (2009)
3. Pushpa, S., Easwarakumar, K.S., Elias, S., Maamar, Z.: Referral based expertise search system in a time evolving social network. In: Proceedings of the Third Annual ACM Bangalore Conference, COMPUTE 2010, pp. 6:1– 6:8. ACM, New York (2010)
4. Yu, B., Singh, M.P.: Searching social networks. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, pp. 65–72. ACM, New York (2003)
5. dos Reis, J.C., Bonacin, R., Baranauskas, M.C.C.: Beyond the Social Search: Personalizing the Semantic Search in Social Networks. In: Ozok, A.A., Zaphiris, P. (eds.) OCSC 2011. LNCS, vol. 6778, pp. 345–354. Springer, Heidelberg (2011)
6. Yeh, J.-H., Sie, S.-H.: Towards Automatic Concept Hierarchy Generation for Specific Knowledge Network. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 982–989. Springer, Heidelberg (2006)

7. Kuo, H.C., Lai, H.C., Huang, J.P.: Building a concept hierarchy automatically and its measuring. In: 2008 International Conference on Machine Learning and Cybernetics, vol. 7, pp. 3975–3978 (2008)
8. Qiao, S., Chunhui, Z., Zhibo, C.: Automatic construction of domain concept hierarchy. In: 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 433–436 (2010)
9. Sun, A., Lim, E.P.: Hierarchical text classification and evaluation. In: Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM 2001, pp. 521–528. IEEE Computer Society, Washington, DC (2001)
10. Gelbukh, A., Sidorov, G., Guzmán-Arenas, A.: Document indexing with a concept hierarchy. In: In New Developments in Digital Libraries. In: Proceedings of the 1st International Workshop on New Developments in Digital Libraries (NDDL 2001). ICEIS Press, Setubal (2001)
11. Chein, M., Mugnier, M.L.: Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs, 1st edn. Springer Publishing Company (2009) (Incorporated)
12. Cilibrasi, R., Vitanyi, P.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19, 370–383 (2007)
13. Kiritchenko, S., Matwin, S., Famili, A.F.: Functional annotation of genes using hierarchical text categorization. In: BioLINK SIG: Linking Literature, Information and Knowledge for Biology (2005)

A Simple Adaptive Algorithm for Numerical Optimization

Francisco Viveros-Jiménez[1], Jose A. León-Borges[2], and Nareli Cruz-Cortés[1]

[1] Centro de Investigación en Computación, Instituto Politécnico Nacional
México D.F.,
[2] Universidad Politécnica de Quintana Roo, México
pacovj@hotmail.com, jleon@upqroo.edu.mx, nareli@cic.ipn.mx

Abstract. This paper describes a novel algorithm for numerical optimization, which we call Simple Adaptive Climbing (**SAC**). SAC is a simple efficient single-point approach that does not require a careful fine-tunning of its two parameters. Our algorithm has a close resemblance to local optimization heuristics such as random walk, gradient descent and, hill-climbing. However, SAC algorithm is capable of performing global optimization efficiently in any kind of space. Tested on 15 well-known unconstrained optimization problems, it confirmed that SAC is competitive against representative state-of-the-art approaches.

1 Introduction

Global optimization is the task of finding the point x^* with the smallest (minimization case) or biggest (maximization case) function value $f(x^*)$. In general, global optimization is a complex task that has remained unsolved until now. Optimizers are efficient tools used for a wide range of tasks such as: file compression [14], scheduling of an aircraft's engine maintenance [9], optimization of financial portfolios [20], evolution of neural networks for walking robots [16], among many others. Most of the recent efficient optimizers for solving unconstrained nonlinear optimization, such as restart CMA-ES [8], can be considered complex approaches because they use the Hessian and covariance matrix, which, in most cases, are very expensive to obtain. Those methods are very effective and greatly overcome simple heuristic approaches [11] [21].

However, we need to close the performance gap between simple heuristics and more complex approaches. This paper, presents a novel approach based on the idea of simplicity: Simple Adaptive Climbing (**SAC**). SAC is a single-point approach similar to hill-climbing or random walk algorithms.

The contents of this paper are organized as follows: First, we give a brief introduction to hill-climbing algorithm in Section 2. Section 3 describes the SAC-algorithm. Section 4 discusses the experimental design. Section 5 presents a comparison of SAC against four state-of-the-art approaches: Elitist Evolution, micro-Particle Swarm Optimization, Simple Adaptive Differential Evolution and Restart CMA-ES. Finally, Section 6 draws some conclusions.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 115–126, 2013.

2 Brief Review of the Hill Climbing-Like Algorithms

A hill-climbing like algorithm performs its search by moving an explorer (single point) through a mountain having a really thick fog (the target function). The explorer performs jumps for landing to a new point. If the new point is better, then the explorer stays in the new position. If the new point is not better, the explorer returns to its previous position and tries to perform a smaller jump. The jump range is known as step size and represents the current algorithm search ratio. This way, the explorer continues jumping until the jump range diminishes.

Hill-climbing like techniques are known for being fast on optimizing local optimum values, i.e. they are very good on descending/ascending a single hill function (unimodal functions). However, this strength is also its greater flaw: they could fail solving functions with multiple hills (multimodal funcions). Three main reasons have been detected behind this behavior [2]:

1. The foothill problem: the algorithm gets stuck in a local optimum, i.e. an optimum value that is not the global one.
2. The plateau problem: the algorithm gets stuck in mostly flat surfaces with few sharp peaks.
3. The ridge problem: the algorithm gets stuck because the ascent direction is not within the possible search directions.

On the other hand, population-based algorithms such as Differential Evolution and Genetic Algorithms have the capability of easily finding optimal value regions. However, they have a relatively slow local optimization speed. For this reason, hill-climbing algorithms are commonly used in combination with population algorithms to create efficient memetic algorithms [3,7].

3 Simple Adaptive Climbing

SAC is a single-point optimizer designed to be simple and competitive. The main idea is to modify the step size according to the current state of the search space. Further, a reinitialization process is added. Particularly, SAC general ideas are: (1) to increase search ratio while improving the solution, (2) to decrease search ratio when no improvements are made, and, (3) to restart the process when no improvements are made for a predetermined number of trials. The algorithm 1 shows the SAC technique for minimization. SAC requires the configuration of two parameters:

1. Base step sizes ($\mathbf{B} \in [\mathbf{0.0}, \mathbf{0.5}]$): B is the search space percentage to be explored in both directions of each dimension. For example, if B = 0.5 means that SAC will search half of the search space to the right and half to the left, meaning the whole search space to be explored. Figure 1 shows an example of the B search ratio in a 2D function.
2. Maximum consecutive errors ($\mathbf{R} > \mathbf{0}$): indicates the maximum consecutive errors necessary to restart the search process. Figure 3 shows an example of a restart in a 2D function.

SAC search by performing alterations in a random dimension subset as shown in Figure 2. It uses adaptive step sizes (as shown in lines 4-12 of Algorithm 1 where $b_j, j = 1, \ldots, D$, and D is the dimensionality of the problem) that is the key of exploration process. SAC adjusts its size accordingly to the current success/failure of the search:

- When a best point is found, b values became greater to encourage exploration of new search areas (see line 15 on Algorithm 1 and Figure 5 for an example).
- When no best point is found, b values became equal to the change between current and former points, encouraging exploration of nearby areas (see line 22 on Algorithm 1 and Figure 6 for an example).

In SAC, the search space is considered as a cincunference (Figure 4). So, if SAC tries to explore outside a variable bound, let us say the the upper limit, then it will explore a valid region near the variable lower limit. To avoid premature convergence on some functions, SAC keeps track of the consecutive unsuccessful explorations (*restart* variable on Algorithm 1). When *restart* reaches the user-defined limit R, the step-sizes and the current position are restarted as seen in line 24 on Algorithm 1.

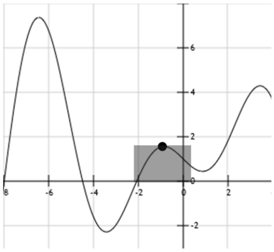


Fig. 1. B represents the initial and maximum search space. A $B = 0.1$ is depicted by the gray rectangle in this picture.

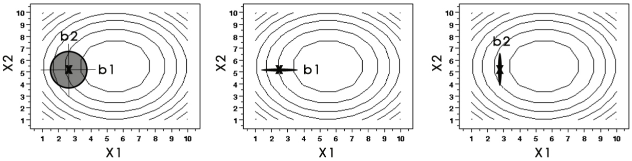


Fig. 2. SAC could search in any dimension subset of this 3D function with a maximum ratio of b_j

```
Data: $B \in [0.0, 0.5]$ (initial step size), $R > 0$ (maximum number of consecutive
      errors) and $MaxFEs$ (maximum number of function evaluations )
Result: $Xbest$ (best solution found)
Set $X$ as a random initial point and $Xbest = X$ as the best known solution;
Set $b_j = B, j = 1, \dots, D$ as the initial step sizes and $restart = 0$;
for $g$=1 To MaxFEs do
    Set $P_a = \frac{rnd(1,D)}{D}$;
    if $flip_j(P_a), j = 1, \dots, D$ then
        Set $O_j = X_j + rndreal(-b_j, b_j) \times (up_j - low_j)$;
        if $O_j > up_j$ then
            Set $O_j = low_j + (O_j - up_j)$;
        if $O_j < low_j$ then
            Set $O_j = up_j - (low_j - O_j)$;
    else
        $O_j = X_j$;
    if $f(O) < f(X)$ then
        Set $restart = 0$;
        Set $b_j = rndreal(b_j, B)$ for all $j$ dimensions where $O_j \neq X_j$;
        Set $X = O$;
        if $f(O) < f(Xbest)$ then
            Set $Xbest = O$;
    else
        Set $restart = restart + 1$;
        if $restart < R$ then
            Set $b_j = \left|\frac{X_j - O_j}{up_j - low_j}\right|$ for all $j$ dimensions where $O_j \neq X_j$;
        else
            Set $restart = 0$, $X = O$ and $b_j = B, j = 1, \dots, D$;
```

Algorithm 1. Algorithm for SAC (minimization case)

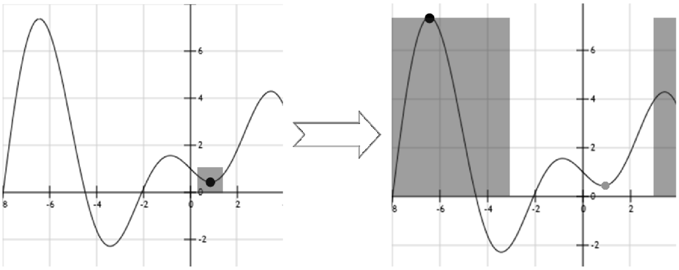


Fig. 3. Position and step sizes are restarted when SAC reaches R iterations without improving the solution. SAC keeps memory of the best solution found (represented as a gray dot in the second picture).

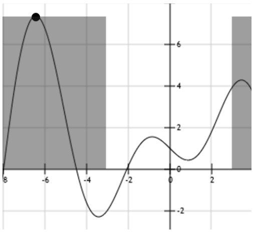


Fig. 4. Function are over a circunference in SAC, i.e., if SAC tries to explore outside the lower limit it will explore a valid region near the upper limit

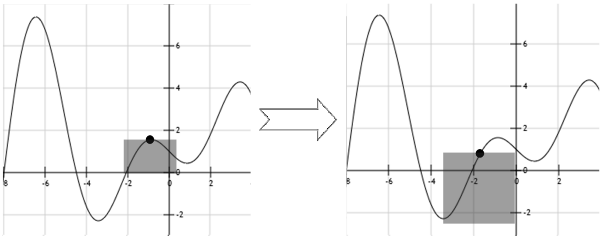


Fig. 5. When improving SAC increases the step sizes

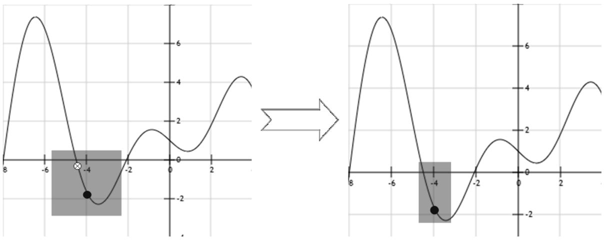


Fig. 6. When failing (white dot) SAC decreases the step sizes

Table 1. Test functions [12] [15]

Unimodal functions	Multimodal functions
Separable	
f_{sph} Sphere model	f_{sch} Generalized Schwefel's problem 2.26
$f_{2.22}$ Schwefel's problem 2.22	f_{ras} Generalized Rastrigin's function
$f_{2.21}$ Schwefel's problem 2.21	
f_{stp} Step function	
f_{qtc} Quartic function	
Non-separable	
$f_{1.2}$ Schwefel's problem 1.2	f_{ros} Generalized Rosenbrock's function
	f_{ack} Ackley's function
	f_{grw} Generalized Griewank's function
	f_{sal} Salomon's function
	f_{whi} Whitley's function
	$f_{pen1,2}$ Generalized penalized functions

4 Experimental Setup

Tests aim to confirm that the proposed algorithm is competitive in solving unconstrained functions. A comparison against some state-of-the-art was performed. The Error and Evaluation values for each trial was measured in a similar way to the one proposed in the test suite for CEC 2005 special session on real-parameter optimization [10]:

- $Error = F(x^o) - F(x^*)$, where x^o is the best reported solution for the corresponding algorithm and x^* is the global optimum value.
- *Evaluation* is the number of function evaluations (**FE**s) required to reach an error value of 10^{-8}.

The benchmark functions are specified in table 1. Thirty trials per test function were conducted and the number of successful trials that reached the target accuracy value were measured. The stop condition criterion for all approaches was $MaxFEs = 3E + 5$ function evaluations (**FE**s). The results were compared against two micro-EAs and two state-of the-art-approaches. Micro-population algorithms were selected because they are considered in between hill-climbers algorithms and other population algorithms. Also, they are competitive against their standard counterparts [1] [6] and are used to create memetic algorithms also [5] [17]. The selected approaches are:

- Elitist Evolution (**EEv**) [19]: The best micro-population algorithm as far as the authors knowledge.
- Micro Particle Swarm Optimization (**μ-PSO**) [13]: The micro population counterpart of PSO, that maintains the original algorithm performance.
- Simple Adaptive Differential Evolution (**SaDE**) [18] selected because it is a competitive Differential Evolution [4] variant representative of the state-of-the-art techniques.

– Restart CMA-ES [8] selected for measuring the gap against a technique that uses Hessian and covariance matrices. This was also the best technique on CEC 2005 special session on real-parameter optimization.

All the experiments were performed using a Pentium 4 PC with 512 MB of RAM, in C language over a Linux environment. The parameter sets for the techniques were:

1. **SAC**: $R = 150, B = 0.5$.
2. **μ-PSO**: $P = 6, C_1 = C_2 = 1.8, Neighborhoods = 2$, Replacement generation = 100, Replacement particles= 2, Mutation % = 10, based on [13].
3. **EEv**: $P = 5, B = 0.5$
4. **SADE**: set as in [18].
5. **Restart CMA-ES**: set as in [8].

5 Test Results and Analysis

5.1 Performance Evaluation on Functions with 30 Dimensions

A comparison of SAC against EEv, was performed μ-PSO, SADE and Restart CMA-ES. Tables 3 and Figure 7 show more detailed test results. Table 2 gives statistical significance to test results. Tests allow us to confirm the following:

1. SAC has a competitive performance in global optimization problems with a high dimensionality without requiring the fine-tuning of its two parameters.
2. SAC outperforms EEv and μ-PSO on most functions. It have more success rate and overall speed.
3. The difference between SAC's performance and EEv and μ-PSO' performances is statistical significant on most functions.
4. SAC is competitive against a state-of-the-art approach like SADE.
5. SAC's success rates competitively against CMA-ES'. However, CMA-ES is faster than SAC.

Table 2. Results of Mann-Whitney U paired test between SAC and other techniques. + means that the improvement is statistically significant. − means that the other technique outperforms SAC. = means that the difference is not statistically significant.

	μ-PSO	EEv	CMA-ES	SADE		μ-PSO	EEv	CMA-ES	SADE
f_{sph}	$+$	$+$	$-$	$=$	f_{ros}	$=$	$=$	$-$	$-$
$f_{2.22}$	$+$	$+$	$-$	$-$	f_{ack}	$+$	$+$	$-$	$=$
$f_{2.21}$	$+$	$+$	$-$	$+$	f_{grw}	$=$	$=$	$-$	$-$
f_{stp}	$+$	$=$	$-$	$-$	f_{pen1}	$+$	$+$	$-$	$+$
f_{qtc}	$+$	$+$	$-$	$+$	f_{pen2}	$+$	$+$	$+$	$=$
f_{sch}	$+$	$+$	$+$	$+$	f_{sal}	$=$	$=$	$=$	$=$
f_{ras}	$+$	$+$	$+$	$+$	f_{whit}	$+$	$+$	$+$	$+$
$f_{1.2}$	$=$	$=$	$-$	$-$					

Table 3. Mean Error values obtained on functions with $D = 30$. A $\star$ value means that 10^{-8} was reached in all runs (100% success rate). On values like X.XE+X(Y) Y represents the success rate (only when $Y \in [1\%, 99\%]$).

$D = 30$	SAC	μ-PSO	EEv	CMA-ES	SaDE
f_{sph}	$\star$	$\star$	$\star$	$\star$	$\star$
$f_{2.22}$	$\star$	$\star$	$\star$	$\star$	$\star$
$f_{2.21}$	$\star$	1.3E-2	9.1E-3	$\star$	4.5E+0
f_{stp}	$\star$	$\star$	$\star$	$\star$	$\star$
f_{qtc}	$\star$	$\star$	$\star$	$\star$	$\star$
$f_{1.2}$	3.2E-4	1.9E-2	6.1E-3	$\star$	$\star$
f_{sch}	4.8E-7(96%)	1.3E+3	1.5E+3	1.2E+4	3.9E+0(96%)
f_{ras}	$\star$	8.1E+0	$\star$	3.3E+0(10%)	7.9E-1(63%)
f_{ros}	1.3E+1	1.6E+1	4.1E+1	$\star$	3.9E-1(63%)
f_{ack}	$\star$	$\star$	$\star$	$\star$	3.1E-2(96%)
f_{grw}	2.3E-2(33%)	2.3E-2(30%)	2.6E-2(23%)	$\star$	2.7E-3(83%)
f_{pen1}	$\star$	$\star$	$\star$	$\star$	6.9E-3(93%)
f_{pen2}	$\star$	$\star$	$\star$	1.4E-3(86%)	$\star$
f_{sal}	6.6E-1	4.5E-1	6.3E-1	2.1E-1	2.0E-1
f_{whit}	6.9E+0(43%)	1.0E+2	1.0+1(40%)	4.8E+2	5.9E+1(20%)

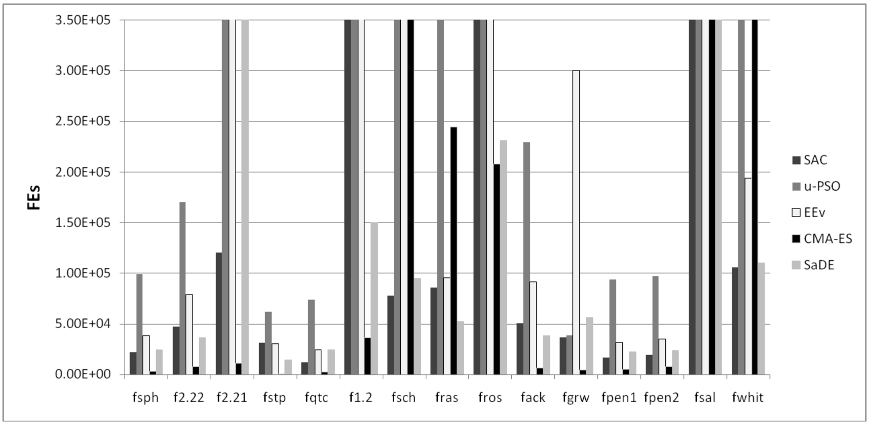


Fig. 7. Speed comparison (Evaluation values) on 30D functions

5.2 Sensibility to Parameter Adjustments

Tests with different B and R values were made to observe the effects of parameter configuration on SAC's performance. Results are shown in Figures 8 and 8, and, Tables 4 and 5. Tests allow observing that:

- SAC does not need complex configuration of its parameters.
- B is the reference search ratio of SAC. Greater B values encourage global exploration and smaller values encourage local exploitation.

Table 4. Changes in the success rates when using different B values. Only the functions with different success rates are shown. Best results are obtained with a $B = 0.5$.

$B = 0.1$		0.2	0.3	0.4	0.5
f_{sch}	–	1%	76	86	96
f_{ras}	–	63	90	93	100
f_{grw}	20	16	13	36	10
f_{whit}	10	23	43	26	43

Table 5. Changes in the success rates when using different R values. Only the functions with different success rates are shown.

$R = 15$		50	150	300	500	$R = 15$		50	150	300	500
f_{sph}	—	100%	100	100	100	f_{ros}	—	3	—	3	3
$f_{2.22}$	—	—	100	100	100	f_{ack}	—	—	100	100	100
$f_{2.21}$	—	—	100	100	100	f_{grw}	—	30	10	30	30
f_{stp}	—	100	100	100	100	f_{pen1}	—	100	100	100	100
f_{qtc}	—	100	100	100	100	f_{pen2}	—	100	100	100	100
f_{sch}	—	60	96	96	90	f_{whit}	—	46	43	46	100
f_{ras}	—	100	100	96	93						

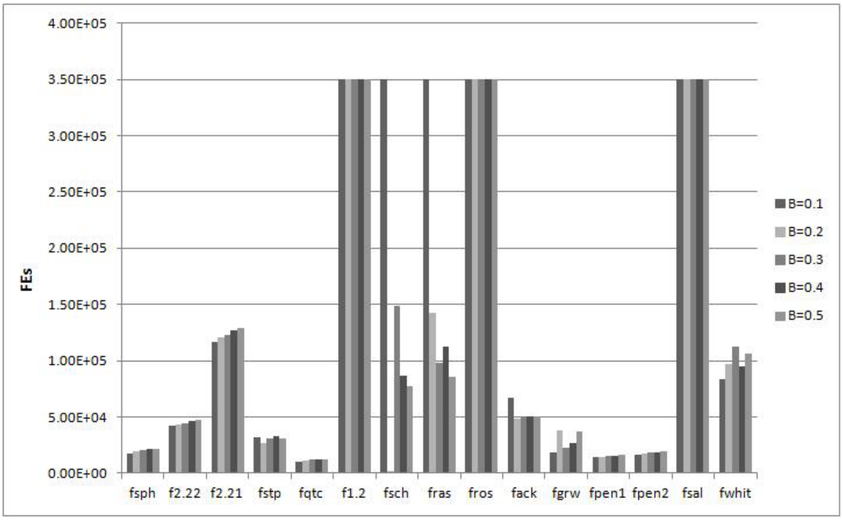


Fig. 8. Evaluation values registered with different B values on 30D functions

- R controls SAC's searching time over the current location. Smaller values allow SAC to move often, while bigger values provoke best exploitation of the current area.
- Smaller B values increase SAC performance on unimodal problems and decrease performance on the multimodal ones (SAC behaves more like a hill-climbing algorithm).

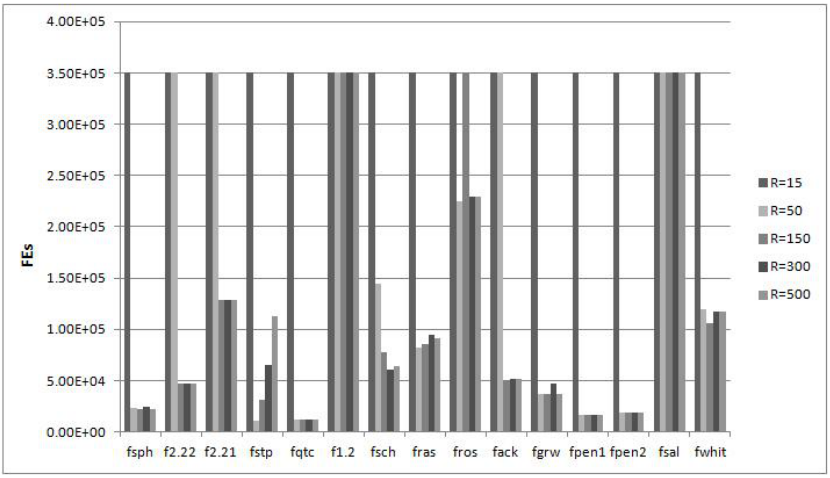


Fig. 9. Evaluation values registered with different R values on 30D functions

- Bigger B values have the best overall performance.
- Smaller R values have poor performance.
- Bigger R values have best performance on multimodal non-separable problems.

6 Conclusions and Future Work

This paper presented a novel optimizer called simple adaptive climbing ((SAC)) and tested it on 15 benchmark functions. SAC is a simple technique that uses a single exploration point and adaptive step sizes. Its main features are: (1) easy implementation, (2) state-of-the-art performance competitive against techniques such as: μ-PSO and SADE, (3) easy parameter configuration, (4) fast solution speed, and, (5) high success rate.

SAC uses 2 parameters: B and R. B is reference search ratio of SAC. Big B values encourage global exploration and small B values encourage local exploitation. R controls SAC's searching time over the current location. Small R values allow SAC to move often, while bigger values necessitate a more extensive exploration over the current search area.

More comparative studies and further analysis should be carried out to provide a more detailed understanding of SAC. It is planned to test SAC in constrained and in multi-objective optimization problems.

Acknowledgments. The first author acknowledges support from CONACyT through a scholarship. The authors acknowledge support from CONACyT through the project 132073 and SIP-IPN 20120002.

References

1. Krishnakumar, K.: Micro-genetic algorithms for stationary and non-stationary function optimization. In: SPIE: Intelligent Control and Adaptive Systems, vol. 1196, pp. 289–296 (1989)
2. Winston, P.H.: Artificial Intelligence, 3rd edn. Addison-Wesley Publishing Company, Reading (1992)
3. Renders, J.M., Bersini, H.: Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 312–317 (1994)
4. Storn, R., Price, K.: Differential Evolution - a simple and efficient heuristic for global optimization. Journal of Global Optimization 11(4), 341–359 (1997)
5. Kazarlis, S.E., Papadakis, S.E., Theocharis, J.B., Petridis, V.: Microgenetic Algorithms as Generalized Hill-Climbing Operators for GA Optimization. Evol. Comput. 5(3), 204–217 (2001)
6. Toscano-Pulido, G., Coello-Coello, C.A.: Multiobjective Optimization using a Micro-Genetic Algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 126–140. Springer, Heidelberg (2001)
7. Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-Coded Memetic Algorithms with Crossover Hill-Climbing. Evolutionary Computation 12(3), 273–302 (2004)
8. Auger, A., Kern, S., Hansen, N.: A Restart CMA Evolution Strategy with Increasing Population Size. In: CEC 2005 Special Session on Real-Parameter Obtimization, Nanyang Technol. Univ., Singaporem IIT Kanpur (2005)
9. Kleeman, M.P., Lamont, G.B.: Solving the Aircraft Engine Maintenance Scheduling Problem Using a Multi-objective Evolutionary Algorithm. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 782–796. Springer, Heidelberg (2005)
10. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Obtimization. Nanyang Technol. Univ, Singaporem IIT Kanpur, 2005005 (2005)
11. Hansen, N.: Compilation of Results on the 2005 CEC Benchmark Function Set. Technical Report, CoLAB Institute of Computational Science, ETH, Zurich (2006)
12. Mezura-Montes, E., Coello Coello, C.A., Velazquez, R.J.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 485–492 (2006)
13. Fuentes Cabrera, J.C., Coello Coello, C.A.: Handling Constraints in Particle Swarm Optimization Using a Small Population Size. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 41–51. Springer, Heidelberg (2007)
14. Kattan, A., Poli, R.: Evolutionary lossless compression with GP-ZIP*. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 1211–1218. ACM (2008)
15. Noman, N., Iba, H.: Accelerating Differential Evolution Using an Adaptive Local Search. IEEE Transactions on Evol. Comput. 12(1), 107–125 (2008)
16. Valsalam, V.K., Miikkulainen, R.: Modular neuroevolution for multilegged locomotion. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 265–272. ACM (2008)

17. Parsopoulos, K.E.: Cooperative Micro-Particle Swarm Optimization. In: ACM 2009 World Summit on Genetic and Evolutionary Computation (2009 GEC Summit), pp. 467–474. ACM, Shanghai (2009)
18. Qin, K.A., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. IEEE Transactions on Evol. Comput. 13(2), 398–417 (2009)
19. Viveros Jiménez, F., Mezura-Montes, E., Gelbukh, A.: Elitistic Evolution: An Efficient Heuristic for Global Optimization. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 171–182. Springer, Heidelberg (2009)
20. Yan, W., Sewell, M.V., Clack, C.D.: Learning to optimize profits beats predicting returns -: comparing techniques for financial portfolio optimisation. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO 2008, pp. 1681–1688. ACM (2009)
21. Hansen, N., Auger, A., Finck, S., Ros, R.: Comparison tables: BBOB 2009 function testbed in 20-D. Technical Report, INRIA (2010)

A Post-optimization Strategy for Combinatorial Testing: Test Suite Reduction through the Identification of Wild Cards and Merge of Rows

Loreto Gonzalez-Hernandez[1], Jose Torres-Jimenez[1], Nelson Rangel-Valdez[2], and Josue Bracho-Rios[1]

[1] Information Technology Laboratory, CINVESTAV-Tamaulipas
Km. 5.5 Carretera Cd. Victoria-Soto la Marina, 87130, Cd. Victoria Tamps., Mexico
[2] Polytechnic University of Victoria
Km. 5.5 Carretera Cd. Victoria-Soto la Marina, 87138, Cd. Victoria Tamps., Mexico
agonzalez@tamps.cinvestav.mx, jtj@cinvestav.mx, nrangelv@upv.edu.mx,
jbracho@tamps.cinvestav.mx
http://www.tamps.cinvestav.mx/~agonzalez/
http://www.tamps.cinvestav.mx/~jtj/

Abstract. The development of a new software system involves extensive tests on the software functionality in order to identify possible failures. It will be ideal to test all possible input cases (configurations), but the exhaustive approach usually demands too large cost and time. The test suite reduction problem can be defined as the task of generating small set of test cases under certain requirements. A way to design test suites is through interaction testing using a matrix called Covering Array, CA$(N; t, k, v)$, which guarantees that all configurations among every t parameters are covered. This paper presents a simple strategy that reduces the number of rows of a CA. The algorithms represent a post-optimization process which detects wild cards (values that can be changed arbitrarily without the CA losses its degree of coverage) and uses them to merge rows. In the experiment, 667 CAs, created by a state-of-the-art algorithm, were subject to the reduction process. The results report a reduction in the size of 347 CAs (52% of the cases). As part of these results, we report the matrix for $CA(42; 2, 8, 6)$ constructed from $CA(57; 2, 8, 6)$ with an impressive reduction of 15 rows, which is the best upper bound so far.

Keywords: Combinatorial testing, Test suite reduction, Covering arrays, Wild cards, IPOG-F.

1 Introduction

Software systems are widely used in our society and they are present in such daily activities as social networking and mobile devices; furthermore they take an important role in scientific and technological development. Therefore, the

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 127–138, 2013.

quality of life of our society is greatly influenced by the software reliability; on the contrary, software failures can cause large losses in the economy [22] or even affect the health or life of people, as stipulated in the records of the Therac 25 crash [18] and the failure of the Ariane 5 rocket [19].

The quality of a software system is highly related to software testing [12]. Software testing has the aim to detect existing defects in a software system, such that the bugs can be corrected before it begins to be used.Various kinds of techniques and methods to ensure software quality have been developed in order to detect different types of failures, one classification involves the *white-box* and the *black-box* test strategies [4]. The white-box approach uses an internal perspective of the software. This approach requires that the tester has programming skills to identify all execution paths through the source code. In contrast, black-box is a functional testing technique, it takes an external perspective of the test object to derive test cases. Taking into account the input configuration, the tester determines if the output is the correct, i.e. the software component must correctly process the input data and provides the expected output depending on the specific task that it performs. In this paper, the term software testing is referred to black-box testing.

During software testing each test case indicates a configuration. In this context, a software system is constituted by components that contain a set of k parameters. A parameter is defined as an element that serves as software input, receiving an unique value from a set of v possible values; therefore each software component has v^k possible configurations. A configuration indicates the setting values for each of the k parameters to execute a test case. It will be ideal to test all input cases; however the exhaustive approach is usually infeasible in terms of time and budget because if the number of parameters increases, the number of configuration grows exponentially. Due to this reason, another alternative to create an effective test suite has to be used.

The test suite reduction problem can be defined as the task of generating a set of test cases, as small as possible, under certain requirements that must be satisfied to provide the desired testing coverage of a system. A way to design test suites is through interaction testing (also called Combinatorial Testing), in which a matrix that involves all the possible combination of symbols that the factors of a system can take, under a certain interaction level.

Combinatorial Testing (CT) is an alternative that can be used for software testing which has been widely applied to construct different instances [5,11]. CT is based on empirical results of different kinds of software which indicate that all their identified failures were triggered by unexpected interactions among at most 6 parameters [14,15]. Based on this premise, CT implements a model-based testing approach using combinatorial objects where a covering array (CA) is one of the most used. A $\mathrm{CA}(N; t, k, v)$ is an $N \times k$ matrix that contains the test suite for the software under test. Every row indicates a test case and the columns represent the k parameters of the software, which can take v symbols. CAs offer a level of coverage t (strength), since every $N \times t$ sub-array includes, at least once, all the ordered subsets from v symbols of size t.

Even a CA significantly reduces the number of test cases, the problem to construct optimal CAs, also known in the literature as *Covering Array Construction*, is consider highly combinatorial [6].

Due to the complexity of the CAC, several techniques have been implemented although they do not necessarily provide the optimal number of rows. Among these strategies are: a) algebraic methods [13,7], b) recursive methods [9,8], c) greedy methods [10,3], d) metaheuristics [23,24] and e) exact methods [1]. A most detailed explanation of all these approaches can be found in recent surveys [16,17].

Among the techniques that provide the fastest results are greedy methods, being one of the best known the IPOG-F algorithm (In-Parameter-Order General). IPOG-F is the primary algorithm used in ACTS tool (Advanced Combinatorial Testing System). It was developed by the NIST, an agency of the United States Government that works to develop tests, test methodologies, and assurance methods; and which within its programs includes a research committee allocated to CT[1].

A post optimization process can be developed through the identification of symbols that are unnecessary in the test suite already constructed, those symbols (which will be named as wild cards), can be substituted by any other one without affecting the coverage of the matrix. In this paper we present a post-optimization process to reduce the size of CAs. This test suite reduction uses two algorithms refereed as `wcCA` and `FastRedu`. The algorithm `wcCA` detects wild cards (symbols that can be changed arbitrarily such that a CA does not lose its level of coverage). If the number of wild cards is greater than zero, `FastRedu` tries to merge compatible rows (two rows are compatible if for each column the two symbols involved are identical or one of them is a wild card) this action allows the reduction of rows from the original CA.

Test suite reduction has been widely studied by several researchers [2,21]; however, the proposed techniques are primary focused on test suites constructed by approaches different to CT, so the features of those test suites do not include the level of coverage of a CA which indicates that every v^t tuples appear in the $\binom{k}{t}$ combinations of parameters at least once. To our knowledge, the only other work that advocates to reduced the size of a CA using a post-optimization process is presented by Colbourn [20].

To test the performance of this approach, the algorithms were tested using 667 CAs created by the deterministic algorithm IPOG-F and obtained from the NIST website [2].

This paper is organized in the following Sections. Section 2 presents the definitions of CA and wild cards, Section 3 provides an explanation of one technique that has been used for post-optimization process of CAs. Section 4 gives an overview of the proposed algorithms `wcCA` and `FastRedu`. After that, Section 5 shows the design of the experiment and the results. Finally, Section 6 summarizes the main contribution of this paper.

[1] http://csrc.nist.gov/groups/SNS/acts/index.html

[2] http://math.nist.gov/coveringarrays/

2 Background of Covering Arrays

2.1 Definition of CA

A CA, denoted by CA$(N; t, k, v)$, is a matrix of size $N \times k$ and strength t where each column has v distinct symbols; and every $N \times t$ sub-array contains all combinations of v^t symbols at least once. When a CA is used for software testing, every column indicates the corresponding parameter of the software under testing and the symbols in the column specify the values for such parameter. Each row represents a test case, i.e. the configuration for an experimental run. A CA is optimal if it has the smallest possible number of rows, the value of N is known as the *Covering Array Number* and is formally defined as

$$\mathrm{CAN}(t, k, v) = min\{N | \exists\ \mathrm{CA}(N; t, k, v)\}$$

To illustrate the use of CAs suppose that we have a system with 3 parameters each with 2 possible values labeled as 0 and 1 respectively as shown in Table 1.

Table 1. System with 3 parameters each with 2 possible values

	O.S.	**Web browser**	**Database**
0 →	Linux	Mozilla Firefox	MySQL
1 →	Windows	Internet Explorer	Oracle

The exhaustive approach demands $2^3 = 8$ configurations, but instead of this, we can use a CA with $t = 2$, i.e. it covers all configurations between pairs of parameters, thus we only need 4 test cases as shown in Table 2. Every row indicates the configuration of a test case.

Table 2. Mapping of the CA(4;2,3,2) to the corresponding pair-wise test suite

	O.S.	**Web browser**	**Database**
1 1 0 →	Linux	Internet Explorer	MySQL
1 0 1 →	Linux	Mozilla Firefox	Oracle
0 1 1 →	Windows	Internet Explorer	Oracle
0 0 0 →	Windows	Mozilla Firefox	MySQL

In the example shown in Table 2 the total of combinations between pairs of parameters is $\binom{k}{t} = \binom{3}{2} = 3$ being these {O.S., Web browser}, {O.S., Database} and {Web browser, Database}. Each parameter has 2 settings giving 4 possible combinations for each pair of them. The definition of a CA implies that every

$N \times t$ sub-array contains all possible combinations of v^t symbols *at least once*, based on this fact, for the tuple {O.S., Web browser} all the combinations {0,0}, {0,1}, {1,0}, {1,1} (mapped to the corresponding settings) are covered, the same way for any pair of selected parameters.

2.2 Detection of Unnecessary Symbols (Wild Cards) in a CA

Within the definition of CA, the indication *at least once* means that a combination can be covered *more than once*, i.e. there is the possibility that some symbols be changed arbitrarily without the CA losses its degree of coverage. These symbols are referred as wild cards. They are exemplified in Table 3 using asterisks *.

Table 3. Detection of wild cards in the CA(5;2,3,2)

A	**B**	**C**		**A**	**B**	**C**
0	0	0		0	0	0
1	0	1		1	0	1
0	1	1	→	0	1	1
1	0	0		*	*	*
1	1	0		1	1	0

The array on the right side is still a CA due to for all pairs of columns {A, B}, {A, C} {B, C} the combinations {0,0}, {0,1}, {1,0}, {1,1} are covered. In this example all symbols in the third row are wild cards, it means that the row can be deleted to obtain a new CA(4;2,3,2) which improves the size of the original.

Wild cards have different applications, one of them is the possibility to merge compatible rows in a CA (reducing its number of rows), in this way, many CAs can be constructed by strategies that demand less time than exact approaches with the possibility of reducing their size through a post-optimization strategy. Another alternative to use wild cards is shown in the work of Colbourn, *et al.* [9] which explains how use them to construct CAs with larger number of columns respect to the input ones through algebraic constructions.

3 Related Work

An algorithm to find wild cards was presented in Nayery *et al.* [20]. The main idea consists in searching elements of the initial array CA($N; t, k, v$) that do not affect coverage of tuples of cardinality t. These elements can be replaced by symbol *, after that, the row containing the greatest number of * is moved to the last place in the array. This number is stored. For all remaining rows containing *, look through all elements of row r where * occurs (let this element occur in column c). If the value in the last row of column c is *, replace element of row r in column c by a random value. Otherwise, replace it by the value in the last row of column c. 8. Permute all rows, except for the last one, in a random way to obtain a new initial array. The detailed explanation of this algorithm is presented in [20].

4 Proposed Approach

The methodology in this paper involves the use of two algorithms. The purpose of the first one is to find wild cards in a given CA using a greedy strategy. The second algorithm reduces the number of rows of the CA resulting of the first step. The next sections describe in detail both algorithms.

4.1 Wild Card Identification Algorithm (wcCA)

This section presents the Wild Card Identification Algorithm (or `wcCA`) for the identification of wild cards. The process to find wild card symbols in a CA$(N; t, k, v)$ is simple and it is described in the following paragraphs.

The algorithm first determines the number of rows that covers the different combinations of symbols for each t-way interaction (a t-way interaction is a combination of t columns of the CA). Whenever a combination of symbols in a t-way interaction is solely covered by a row, the elements of the row involved in the corresponding columns are marked as fixed. Finally, it selects in a greedy way one covering row for those combination of symbols that are covered more than once; the result of the selection will produce that the chosen row will have fixed the columns for that combination of symbols. Those columns that weren't fixed during this process are wild cards of the input matrix. The Algorithm 2 presents the pseudocode for `wcCA`.

The input for `wcCA` is a matrix $\mathcal{M}_{N\times k}$ that is a CA$(N; t, k, v)$. The output is a matrix $\mathcal{M}'_{N\times k}$ where each cell $m'_{i,j}$ is a wild card if it is assigned an `UNFIXED` value. The function `Fixing` identifies which combination of symbols are covered once. The structures $\mathcal{M}_C$ and $\mathcal{M}_R$ refer to the sets of columns and rows of $\mathcal{M}$, respectively. The variable M is a combination of t columns (*tuples*); it is used in combination with the structure $\texttt{solelyCovered}_{nc}$ to keep track of which combination of symbols nc are covered only once (the value of $\texttt{solelyCovered}_{nc}$ contains the row that covers nc). The function `SymbolCombination`$(\mathcal{M}, r, M)$ returns an integer value that identify which combination of symbols is located in the t columns identified by M in the row r. The function `fixSymbols`$(\mathcal{M}', r, M)$ set to the value `FIXED` the set of columns M in the row r. The function `Fixed`$(\mathcal{M}', r)$ returns the number of fixed columns. Finally, the structure $\texttt{coveredBy}_{nc}$ is filled with the result of the greedy criterion to untie combination of symbols and selects one row to cover them. In general, the algorithm `wcCA` can identify wild cards in time $O(N\binom{k}{t})$.

4.2 Reduction Algorithm Using Wild Cards (FastRedu)

The size of a CA can be reduced by merging compatible rows. Two rows are compatible if for each column they have the same symbol or at least one of the

Algorithm 1. Function Fixing, procedure prior to the wcCA Algorithm

```
Fixing($\mathcal{M}_{N\times k}, N, t, k, v$)
Output: $\mathcal{M}'_{N\times k}$
$\mathcal{M}'_{N\times k}$ ← UNFIXED;
solelyCovered$_{v^t}$ ← $\emptyset$;
foreach $\{M | M \subset \mathcal{M}_C, |M| = t\}$ do
    foreach $r \in \mathcal{M}_R$ do
        $v$ ← SymbolCombination($\mathcal{M}, r, M$);
        if solelyCovered$_v = \emptyset$ then
            solelyCovered$_v$ ← $r$;
        end
        else
            solelyCovered$_v$ ← NOT;
        end
    end
    foreach $v \in V^t$ do
        if solelyCovered$_v \neq$ NOT then
            $r$ ← solelyCovered$_v$;
            fixSymbols($\mathcal{M}', r, M$);
        end
    end
end
return $\mathcal{M}'$;
```

Algorithm 2. Algorithm for the identification of wild cards in a CA

```
wcCA($\mathcal{M}_{N\times k}, N, t, k, v$)
Output: $\mathcal{M}'_{N\times k}$
$\mathcal{M}'_{N\times k}$ ← Fixing($\mathcal{M}, N, t, k, v$);
coveredBy$_{v^t}$ ← $\emptyset$;
foreach $\{M | M \subset \mathcal{M}_C, |M| = t\}$ do
    foreach $r \in \mathcal{M}_R$ do
        $v$ ← SymbolCombination($\mathcal{M}, r, M$);
        $r^*$ ← coveredBy$_v$;
        if Fixed($\mathcal{M}', r$) > Fixed($\mathcal{M}', r^*$) then
            coveredBy$_v$ ← $r$;
        end
    end
    foreach $v \in V^t$ do
        $r$ ← coveredBy$_v$;
        fixSymbols($\mathcal{M}', r, M$);
    end
end
return $\mathcal{M}'$;
```

rows has a wild card. Example of compatible rows are shown in Table 4; here, a wild card is identified by an asterisk *. The example presents two rows that are compatible and the row resulting from merging both rows. The row resulting from the merging process will be the one with the greatest number of wild cards.

Algorithm 3 presents the pseudocode for the reduction algorithm (FastRedu). This algorithm is quite simple. It tests every combination of two rows (r_i, r_j) (lines 3 and 4), where $i < j$ and $r_i, r_j \in \mathcal{M}_R$, and verifies if they are compatible (line 5). Whenever a combination of rows (r_i, r_j) is compatible, they are merged in row r_j and the other row is marked as `unnecessary` in the CA (lines 6 and 7). All the rows marked as unnecessary will be deleted.

Table 4. Example of pairs of compatible rows

(a)	(b)	(c)	(d)
1 1 0 1	0 1 1 0 1	0 * * 0 1	* 1 1 0 1
0 1 1 0 1	0 1 1 * 1	0 1 1 0 1	* 1 1 0 1
0 1 1 0 1	0 1 1 * 1	0 * * 0 1	* 1 1 0 1

Algorithm 3. Algorithm to merge compatible rows in order to reduce the size of a CA

```
FastRedu($\mathcal{M}_{N\times k}, N, t, k, v$)
Output: $\mathcal{M}'_{N\times k}$
$\mathcal{M}'_{N\times k} \leftarrow$ wcCA($\mathcal{M}, N, t, k, v$);
$\mathcal{M}'_{N\times k} \leftarrow$ markFixed($\mathcal{M}, \mathcal{M}'$);
for $i = 1$ to $N$ do
    for $j = i + 1$ to $N$ do
        if areCompatible($\mathcal{M}'_{i,*}, \mathcal{M}'_{j,*}$) then
            mergeRows($\mathcal{M}'_{i,*}, \mathcal{M}'_{j,*}$);
            markRow($\mathcal{M}'_{i,unnecesary}$);
        end
    end
end
return $\mathcal{M}'$;
```

The algorithm FastRedu runs in $O(N^2)$ time, where N is the number of rows of the CA.

5 Experimental Design

The wcCA and FastRedu algorithms were implemented in C language and compiled with gcc. The experiment was carried out in a CPU Intel Core 2 Duo at 1.5 GHz, 2 GB of RAM and Ubuntu 8.10 Intrepid Ibex Operating System. The CAs used in the experiment were generated by the deterministic algorithm IPOG-F [10] and obtained from the NIST webpage [3]. The alphabets v of the instances vary from 2 to 6, columns k from 3 to 32 and strengths t from 2 to 6 giving a whole of 667 CAs.

The experiment was conducted using the algorithms wcCA and FastRedu in the following post-optimization process: firstly, we use wcCA to find wild cards in the 667 CAs; after that, the resulted matrices from wcCA are given to FastRedu to merge compatible pairs of rows.

A summary of the main results obtained from the experiment is shown in Table 5. It can be seen more than 52% (347/667) of the input CAs reduced their size through the post-optimization process. Note the increasing trend on the percentage of improvements respect to the strength (t) of the CA; this suggests

[3] http://math.nist.gov/coveringarrays/ipof/ipof-results.html

Table 5. Improved cases over the total of input CAs after the post-optimization process

(a) Improved cases

v	*t=2*	*t=3*	*t=4*	*t=5*	*t=6*	**Total**
2	0/30	3/29	9/28	11/27	20/26	43/140
3	0/30	6/29	14/28	25/27	25/26	70/140
4	2/30	5/29	18/28	26/27	26/26	77/140
5	3/30	13/29	25/28	27/27	19/19	87/133
6	3/30	13/29	27/28	27/27	-	70/114
Total	8/150	40/145	93/140	116/135	90/97	**347/667**

(b) Percentage of improved CAs

v	*t=2*	*t=3*	*t=4*	*t=5*	*t=6*	**Total**
2	0	10.34	32.14	40.74	76.92	30.71
3	0	20.69	50.00	92.59	96.15	50.00
4	6.67	17.24	64.29	96.30	100	55.00
5	10.00	44.83	89.29	100	100	65.41
6	10.00	44.83	96.43	100	-	61.40
Total	5.33	27.59	66.43	85.93	92.78	**52.02**

Table 6. Minimum and maximum spent time (in sec.) for the post-optimization process

(a) Spent time by the algorithm `wcCA`

t	2		3		4		5		6	
v	min	max	min	max	min	max	min	max	min	max
2	0	0.02	0	0.03	0	0.5	0	8.91	0.01	109.89
3	0	0.02	0	0.08	0	2.9	0	79.81	0	1653.1
4	0	0.01	0	0.18	0	9.34	0.01	414.87	0.02	12783.4
5	0	0.03	0.01	0.41	0	25.18	0.01	1658.21	0.06	8615.71
6	0	0.03	0	0.65	0.01	68.64	0.03	4548.92	—	

(b) Spent time by the algorithm `FastRedu`

t	2		3		4		5		6	
v	min	max	min	max	min	max	min	max	min	max
2	0	0	0	0	0	0.01	0	0.01	0	0.03
3	0	0	0	0	0	0.01	0	0.08	0	1.08
4	0	0	0	0.01	0	0.07	0	1.54	0	67.53
5	0	0	0	0.01	0	0.29	0.02	19.18	0.04	384.66
6	0	0	0	0.03	0	1.2	0.03	157.89	—	

that the possibility of decreasing rows in CAs (constructed by IPOG-F) grows along with the value of t.

Table 7 shows an alternative analysis of the results derived from our experiment. In this new analysis we group the CAs by the number of their columns and their strength. Every group of t contains the different values of the alphabet for each CA. Every cell of the this Table shows the number of rows reduced in the corresponding CA. As seen in the last column, the number of improved cases is mostly concentrated in $k \leq 12$.

The results in Table 7 indicate an impressive reduction in the case CA(57;2,8,6), whose size was reduced by 15 rows. These cases are an example of the reduction of size for CAs that can be obtained through an algorithm to merge rows using wild cards (like `wcCA`, presented in this paper) using as input CAs constructed by the deterministic algorithm IPOG-F. The CA(42;2,8,6) which was obtained by this process is shown in Table 8.

Summarizing, the performance of `FastRedu` as a post-optimization algorithm to reduce CAs shows an improvement in the CA size when the value of v and/or the value of t increases, i.e. the greater the values of t or v are, the higher the possibility to reduce rows from a CA generated by IPOG-F.

Table 7. Number of reduced rows for each CA

	t=2					*t=3*					*t=4*					*t=5*					*t=6*				**Improved**
v k	**2**	**3**	**4**	**5**	**6**	**2**	**3**	**4**	**5**	**6**	**2**	**3**	**4**	**5**	**6**	**2**	**3**	**4**	**5**	**6**	**2**	**3**	**4**	**5**	**cases**
3	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
4	0	0	0	1	0	0	0	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4
5	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	9
6	0	0	0	1	0	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	16
7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	16
8	0	0	0	0	15	1	0	0	1	1	1	2	1	1	1	1	3	1	1	1	1	1	1	2	18
9	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	2	16
10	0	0	0	0	0	0	1	1	1	1	1	2	1	1	5	1	1	1	1	1	1	1	2	3	18
11	0	0	0	0	0	0	0	0	0	0	0	2	1	5	1	0	1	1	4	1	1	2	2	5	12
12	0	0	0	1	0	0	1	1	1	1	0	1	2	3	1	1	3	4	2	1	1	5	1	1	18
13	0	0	0	0	1	0	0	0	0	0	0	2	3	1	1	0	1	1	1	1	6	2	6	1	13
14	0	0	0	0	0	0	0	0	0	0	1	1	5	1	1	1	5	1	2	1	1	1	1	3	14
15	0	0	0	0	1	0	0	0	0	1	0	1	2	1	12	0	1	4	1	1	1	1	2	1	14
16	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	6	2	1	1	1	2	3	1	12
17	0	0	0	0	0	0	0	0	0	0	0	0	3	1	1	0	1	1	1	3	1	3	4	1	11
18	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	4	1	1	1	7	2	3	13
19	0	0	0	0	0	0	0	0	1	1	0	1	1	2	1	0	2	2	3	1	1	4	5	2	14
20	0	0	0	0	0	0	0	0	0	1	0	0	0	2	6	1	3	6	5	1	1	8	2	1	12
21	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	2	1	1	0	7	5	1	11
22	0	0	0	0	0	0	0	0	0	1	0	0	0	2	1	0	1	2	6	1	1	3	3	1	11
23	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	3	3	1	1	0	3	15	1	9
24	0	0	1	0	0	0	0	0	0	1	0	0	1	1	2	0	1	1	2	1	0	2	3	1	12
25	0	0	1	0	0	1	0	0	1	0	0	0	0	3	1	1	2	1	4	2	1	1	1	1	14
26	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1	2	0	6	0	5
27	0	0	0	0	0	0	0	0	1	0	0	0	0	2	1	0	1	3	1	1	1	3	2	0	10
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	3	10	13	0	1	1	0	6
29	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1	2	4	4	1	3	2	0	11
30	0	0	0	0	0	0	1	0	0	0	1	0	1	1	2	0	2	1	2	1	1	2	6	0	12
31	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	1	1	1	2	7	0	2	3	0	9
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	2	4	2	0	1	12	0	7
Improved cases	0	0	2	3	3	3	6	5	13	13	9	14	18	25	27	11	25	26	27	27	20	25	26	19	347

Table 8. The CA(42;2,8,6)^T that was created after the reduction of CA(57;2,8,6) using `wcCA` and `FastRedu`

*	*	*	*	*	*	0	2	2	1	1	1	5	2	4	0	4	4	0	0	3	2	4	5	5	5	0	3	5	2	4	4	5	3	0	1	1	2	3	3	3	1
*	*	*	*	*	*	0	5	4	4	3	2	4	2	2	5	0	1	1	2	4	1	3	0	2	3	3	3	1	0	4	5	5	1	4	0	5	3	0	5	2	1
5	0	3	2	4	1	0	1	5	2	1	4	0	0	3	3	5	1	4	2	3	2	0	2	1	4	5	2	3	4	4	2	5	0	1	3	0	3	1	4	5	5
5	4	0	2	3	1	0	4	2	1	0	4	3	1	5	1	1	4	2	3	4	0	2	4	2	1	4	5	5	5	0	3	0	1	5	2	5	3	3	2	0	3
5	4	3	0	2	1	0	0	4	3	4	5	5	2	0	5	4	2	1	4	0	5	5	1	3	0	3	2	4	3	1	3	2	3	2	2	1	1	5	4	1	0
5	4	3	2	1	0	0	5	0	4	2	0	2	3	4	2	1	2	4	5	1	1	3	3	1	5	1	0	0	2	5	0	4	5	3	5	1	4	4	3	2	3
5	4	3	2	0	1	0	0	3	0	5	2	1	5	1	4	2	3	5	3	5	1	0	5	4	3	1	4	0	4	4	5	2	2	2	1	3	2	3	1	0	4
0	4	3	2	5	1	0	3	5	0	5	3	3	2	5	5	3	0	3	1	1	4	1	5	0	4	2	3	2	1	2	4	1	5	4	4	2	0	2	0	4	1

6 Conclusions

We present a post-optimization strategy to reduce the size of a CA. The post-optimization process reduces the number of rows of a CA through the merging of rows. The strategy to merge rows is performed in two steps. The first one consists on identifying wild cards (symbols that can be changed arbitrarily such that a CA does not lose its level of coverage) with `wcCA` algorithm. The second step merges compatible pairs of rows through `FastRedu` algorithm; two rows are compatible if they share the same symbols in each of their columns or at least one of them is a wild card.

The algorithm to identify wild cards (wcCA) runs in $O(N\binom{k}{t})$ steps, where N is the size of the CA, and t is the strength. The algorithm to merge rows runs in time $O(N^2)$.

The post-optimization process was tested with 667 CAs constructed by the state-of-the-art algorithm IPOG-F. The results show a reduction in 52% of the instances. The CA(57;2,8,6) reduced its size by 15 rows, an impressive reduction if we consider that the new CA(42;2,8,6) is the best upper bound so far.

The improved cases were analyzed in terms of t, k, v. The improvement that can be achieved by the FastRedu algorithm increased with the strength t. An slightly small improvement can also be perceived when the alphabet v is increased. With respect to the number of columns k, the best improvements are concentrated in values of $k \leq 12$.

In conclusion, the quality of the CAs generated by IPOG-F can be improved significantly through our approach with a high probability when the values of t and v are large.

References

1. Bracho-Rios, J., Torres-Jimenez, J., Rodriguez-Tello, E.: A New Backtracking Algorithm for Constructing Binary Covering Arrays of Variable Strength. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS, vol. 5845, pp. 397–407. Springer, Heidelberg (2009)
2. Bryce, R.C., Colbourn, C.J.: Prioritized interaction testing for pair-wise coverage with seeding and constraints. Information and Software Technology 48(10), 960–970 (2006)
3. Bryce, R.C., Colbourn, C.J.: The density algorithm for pairwise interaction testing: Research articles. Software Testing, Verification and Reliability 17, 159–182 (2007)
4. Burnstein, I.: Practical software testing: a process-oriented approach. Springer Professional Computing (2003) ISBN: 0-387-95131-8
5. Changhai, N., Hareton, L.: A survey of combinatorial testing. ACM Computing Surveys (CSUR) 43, 11:1–11:29 (2011)
6. Colbourn, C.J.: Combinatorial aspects of covering arrays. Le Matematiche (Catania) 58, 121–167 (2004)
7. Colbourn, C.J.: Covering arrays from cyclotomy. Designs, Codes and Cryptography 55, 201–219 (2010)
8. Colbourn, C.J., Martirosyan, S., Trung, T., Walker II., R.A.: Roux-type constructions for covering arrays of strengths three and four. Designs, Codes and Cryptography 41, 33–57 (2006), doi:10.1007/s10623-006-0020-8
9. Colbourn, C.J., Martirosyan, S.S., Mullen, G.L., Shasha, D., Sherwood, G.B., Yucas, J.L.: Products of mixed covering arrays of strength two. Journal of Combinatorial Designs 14(2), 124–138 (2006)
10. Forbes, M., Lawrence, J., Lei, Y., Kacker, R.N., Kuhn, D.R.: Refining the in-parameter-order strategy for constructing covering arrays. Journal of Research of the National Institute of Standards and Technology 113(5), 287–297 (2008)
11. Gonzalez-Hernandez, L., Rangel-Valdez, N., Torres-Jimenez, J.: Construction of mixed covering arrays of strengths 2 through 6 using a tabu search approach. In: Discrete Mathematics, Algorithms and Applications, pp. 1–20 (to appear, 2012)

12. Hartman, A., Ben, I.: Graph Theory, combinatorics and algorithms interdisciplinary applications, 4th edn. Springer, New York (2005) ISBN 13:9780387243474
13. Ji, L., Yin, J.: Constructions of new orthogonal arrays and covering arrays of strength three. Journal of Combinatorial Theory Series A 117, 236–247 (2010)
14. Kuhn, D.R., Reilly, M.J.: An investigation of the applicability of design of experiments to software testing. In: Proceedings of the 27th Annual NASA Goddard Software Engineering Workshop (SEW-27'02), SEW 2002, pp. 91–95. IEEE Computer Society, Washington, DC (2002)
15. Kuhn, D.R., Wallance, D.R., Gallo Jr., A.M.: Software fault interactions and implications for software testing. IEEE Transactions on Software Engineering 30, 418–421 (2004)
16. Kuliamin, V., Petukhov, A.: A survey of methods for constructing covering arrays. Programming and Computer Software 37, 121–146 (2011)
17. Lawrence, J., Kacker, R.N., Lei, Y., Kuhn, D.R., Forbes, M.: A survey of binary covering arrays. Electronic Journals of Combinatorics 18, 84 (2011)
18. Levenson, N.G., Turner, C.S.: An investigation of the therac-25 accidents. IEEE Computer 26, 18–41 (1993)
19. Lions, J.L.: Ariane 5, flight 501, report of the inquiry board. European Space Agency (July 1996)
20. Nayeri, P., Colbourn, C.J., Konjevod, G.: Randomized Postoptimization of Covering Arrays. In: Fiala, J., Kratochvíl, J., Miller, M. (eds.) IWOCA 2009. LNCS, vol. 5874, pp. 408–419. Springer, Heidelberg (2009)
21. Sampath, S., Bryce, R.C.: Improving the effectiveness of test suite reduction for user-session-based testing of web applications. Information and Software Technology 54(7), 724–738 (2012)
22. Tassey, G.: The economic impacts of inadequate infrastructure for software testing. Technical report, National Institute of Standards and Technology (May 2002)
23. Torres-Jimenez, J., Rangel-Valdez, N., Gonzalez-Hernandez, L.: An exact approach to maximize the number of wild cards in a covering array. In: Submitted in Mexican International Conference on Artificial Intelligence, Puebla, México, November 26-December 4 (2011)
24. Torres-Jimenez, J., Rodriguez-Tello, E.: Simulated annealing for constructing binary covering arrays of variable strength. In: IEEE Congress on Evolutionary Computation CEC 2010, July 18-23, pp. 1–8 (2010)

A New Branch and Bound Algorithm for the Cyclic Bandwidth Problem

Hillel Romero-Monsivais, Eduardo Rodriguez-Tello, and Gabriel Ramírez

CINVESTAV-Tamaulipas, Information Technology Laboratory,
Km. 5.5 Carretera Victoria-Soto La Marina, 87130 Victoria Tamps., Mexico
{hromero,ertello,grtorres}@tamps.cinvestav.mx

Abstract. The *Cyclic Bandwidth* problem (CB) for graphs consists in labeling the vertices of a guest graph G by distinct vertices of a host cycle C (both of order n) in such a way that the maximum distance in the cycle between adjacent vertices in G is minimized. The CB problem arises in application areas like VLSI designs, data structure representations and interconnection networks for parallel computer systems.

In this paper a new *Branch and Bound* (B&B) algorithm for the CB problem is introduced. Its key components were carefully devised after an in-depth analysis of the given problem. The practical effectiveness of this algorithm is shown through extensive experimentation over 20 standard graphs. The results show that the proposed exact algorithm attains the lower bounds for these graphs (of order $n \leq 40$) expending a reasonable computational time.

Keywords: Branch and Bound, Cyclic Bandwidth Problem.

1 Introduction

The *Cyclic Bandwidth* problem (CB) was first stated by Leung *et al.* in 1984 [1] in relation with the design of a ring interconnection network for a set of computers (V) whose communication pattern was described by a graph $G(V, E)$ where $\{u, v\} \in E$ if computer u communicates with computer v. They were interested in finding an arrangement of these computers on a cycle so that every message sent can arrive at its destination in at most k steps. The CB problem arises also in other application areas like VLSI designs [2], data structure representations [3], code design [4] and interconnection networks for parallel computer systems [5].

The CB problem can be formally defined as follows. Let $G(V, E)$ be a finite undirected graph, where V ($|V| = n$) defines the set of vertices and $E \subseteq V \times V$ $= \{\{u, v\} \mid u, v \in V\}$ is the set of edges. Given a bijective labeling function for the vertices of G, $\varphi : V \to \{1, 2, \ldots, n\}$, the cyclic bandwidth (the cost) for G with respect to the labeling φ is defined as:

$$B_c(G, \varphi) = \max_{(u,v)\in E} \{|\varphi(u) - \varphi(v)|_n\}, \tag{1}$$

where $|x|_n = \min\{|x|, n - |x|\}$ for $0 < |x| < n$, which is called *cyclic distance*. Then the CB problem consists in finding a labeling φ^*, which minimizes

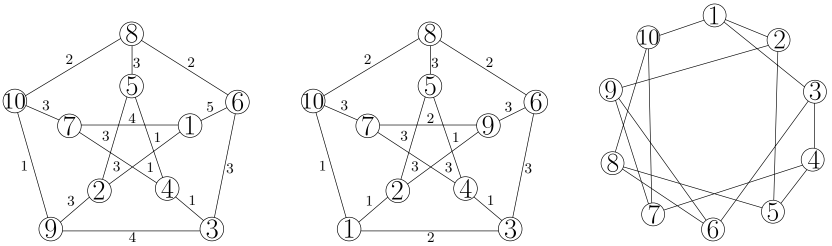
I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 139–150, 2013.

$B_c(G, \varphi^*)$, i.e.,

$$B_c(G, \varphi^*) = \min\{B_c(G, \varphi) : \varphi \in \mathscr{L}\}\,, \tag{2}$$

where $\mathscr{L}$ is the set of all possible labelings.

For instance, consider the graph $G(V, E)$ depicted in Fig. 1(a), consisting of ten vertices with a labeling φ given by the numbers shown inside each vertex. The cyclic distance between each pair of adjacent vertices $(u, v) \in E$ can be calculated using the expression $\min\{|\varphi(u) - \varphi(v)|, n - |\varphi(u) - \varphi(v)|\}$. These cyclic distances are represented by the numbers associated to the edges in the graph. For this particular labeling φ, $B_c(G, \varphi) = 4$. On the other hand, if the labels 1 and 9 are exchanged a new labeling φ' is obtained (see Fig. 1(b)), which produces a smaller (better) cyclic bandwidth value ($B_c(G, \varphi') = 3$). The embedding of this new labeling of the graph G in a cycle is presented in Fig. 1(c) for illustrative purposes.



(a) Labeling φ

(b) Labeling φ'

(c) Embedding G in a cycle

Fig. 1. Example of a *Cyclic Bandwidth* problem instance

It was demonstrated that finding the cyclic bandwidth is NP-complete for general graphs [6]. It is important to note that the CB problem is a natural extension of the well-known *bandwidth minimization* problem for graphs [7], which consists in embedding the vertices of a guest graph G in a host path P (both of order n) in such a way that the maximum distance in the path between adjacent vertices in G is minimized.

In this paper, a new *Branch and Bound* (B&B) algorithm for finding exact solutions for the CB problem for general graphs is presented. The proposed B&B algorithm is based on a lowest-cost search strategy, i.e., branches having the lowest cost are explored while branches with higher cyclic bandwidth are cut. It implements the Hungarian method [8] for solving the problem of efficiently computing the cost of assigning every available label in a partial solution to each possible non-labeled vertex. The practical usefulness of the proposed B&B is assessed experimentally using a test-suite, composed by 20 standard graphs belonging to 4 different classes. The computational results from this experiments show that our algorithm is able to attain the lower bounds for these graphs (of order $n \leq 40$) expending a reasonable computation time.

The remainder of this paper is organized as follows. In Sect. 2 a brief review is given to present some representative related work. The main components of the proposed B&B algorithm are presented in detail in Sect. 3. Section 4 describes the experimental methodology used for assessing the practical usefulness of the proposed B&B algorithm, while the computational results produced by these experiments are presented in Sect. 5. Finally, the conclusions of this work and some possible directions for future research are provided in Sect. 6.

2 Relevant Related Work

In spite of its practical and theoretical importance, less attention has been paid to the CB problem with respect to other graph labeling problems. Up to now, most of the research on this important labeling problem has been devoted to the theoretical study of its properties with the aim of finding exact solutions for certain specific cases. Next, we present a brief review of these studies.

In 2002 Zhou [9] proposed a systematic method for obtaining lower bounds for the bandwidth and cyclic bandwidth problems in terms of some distance- and degree-related parameters of the graph. The main idea of this method is to relax the condition of embedding the graph G on the host graph with the aid of a graphical parameter possessing some kind of monotonic property. This method has been demonstrated to be efficient when the parameters are chosen appropriately. The author concludes that this method yields a number of lower bounds for the ordinary and cyclic bandwidths. In both cases, it gives rise to new estimations, as well as improvements of some known results.

Later, de Klerk *et al.* [10] proposed two new semidefinite programming (SDP) relaxations of the bandwidth and cyclic bandwidth based on the quadratic assignment problem (QAP) reformulation. The bounds produced by this method were tested for some special graphs showing that they are tight for paths, cliques, complete bipartite graphs. However, these bounds are not tight for hypercubes, rectangular grids and complete k-level t-ary trees.

In 1995 Yuan and Zhou [11] demonstrated that for unit interval graphs, there exists a labeling which is simultaneously optimal for the following seven labeling problems: bandwidth, cyclic bandwidth, profile, cutwidth, modified cutwidth and bandwidth sum. Following this idea, in [12] Lam *et al.* made a characterization of graphs with equal bandwidth and cyclic bandwidth which includes planar graphs, triangulation meshes and grids with some specific characteristics.

To the best of our knowledge, there has been no previous research investigating *Branch and Bound* based algorithms for solving the CB problem in the case of general graphs. Therefore, this paper presents an original contribution in this field.

3 A New Branch and Bound Algorithm

In this section, we present a new exact algorithm, based on *Branch and Bound* (B&B), for solving the CB problem. This algorithm employs a lowest-cost search strategy, where branches having the lowest cost are explored while branches with higher cost are cut. The following subsections describe in detail the main components of our algorithm.

3.1 Branching Strategy

Every possible labeling φ_k, for $k = \{1, 2, 3, ..., (n-1)!/2\}$, must be tested. In our algorithm, the labelings are tested in lexicographic order, beginning with $\varphi_1 = \{1, 2, 3, ..., n\}$ and ending with $\varphi_{(n-1)!/2} = \{n, n-1, n-2, ..., 1\}$. Each individual label in φ_k is assigned to vertices, one at a time, producing three different sets: V' the set of already labeled vertices, V''_k the set of unlabeled vertices which are adjacent to those in V' and φ''_k the lexicographic order set of available labels.

The cyclic bandwidth (cost) $B_c(G, \varphi'_k)$ for every particular partial labeling is computed considering V' in G. If all labels in φ_k have been assigned, then a labeling φ'_k with a value $B_c(G, \varphi'_k)$ smaller than the prefixed bound b has been found. The algorithm then stores this new best solution in φ^σ.

According to Leung [1], the upper bound for the CB is $n/2$, so at the beginning $b = n/2$. When a labeling φ^σ is found, the search retakes a new bound $b = B_c(G, \varphi^\sigma) - 1$.

The algorithm performs a depth-first search, so if a certain label is assigned to a vertex and $B_c(G, \varphi'_k) > b$, the branch is cut. Then, a backtrack to the parent label is applied and then the algorithm advances to the next child label. The search process stops when all labelings have been analyzed. A detailed description of the proposed B&B method is summarized in Algorithm 1.

3.2 Assignment Strategy

Two important issues are considered when adding a new label to a partial solution φ'_k: the cost generated by that new label, and the lowest cost which can be produced by assigning the available labels φ''_k to the set V''_k of unlabeled vertices adjacent to those in V'. The latter provides information allowing to know if the adjacent vertices can be labeled without exceeding the current bound b in further iterations.

The Hungarian method is an algorithm which solves the assignment problem in polynomial time [8]. Our implementation employs this algorithm to compute the cost of assigning each label $l \in \varphi''_k$ to each possible vertex $v \in V''_k$. However, if the cost of an assignment becomes higher than the current bound b, then the evaluation of φ_k is not completed.

For instance, consider the partial representation of a graph $G = (V, E)$ depicted in Fig. 2, which consists of five vertices labeled with the number shown within each vertex. Let $n = 10$ and $b = 3$ for this example. Since the vertices in the set V are taken following a lexicographic order, then the last labeled vertex was $V_6 \leftarrow 5$, therefore $V' = \{...V_5, V_6\}$, $\varphi'_k = \{..., 3, 5\}$ and $V''_k = \{V_7, V_8, V_9\}$. Suppose that $\varphi''_k = \{1, 4, 9, 10\}$ and for this partial labeling $B_c(G, \varphi'_k) < b$. Using the Hungarian method the costs of assigning the labels in φ''_k to the vertices in V''_k are estimated and presented in Table 1. From this table it is clear that

Algorithm 1. B&B for the CB problem

```
**B&B**$(G(E, V))$
$b = \frac{n}{2}$
$i \leftarrow 1$
$\varphi_i = \{1, 2, 3, ..., n\}$
$V' \leftarrow \varphi' \leftarrow \varphi^\sigma \leftarrow \emptyset$
while $i \leq (n-1)!/2$ do
    $V_i \leftarrow \varphi_i$
    $\varphi' \leftarrow \varphi' \cup \{\varphi_i\}$
    $V' \leftarrow V' \cup \{V_i\}$
    $c \leftarrow B_c(G, \varphi')$
    if $c > b$ then
        $j \leftarrow$ backtracking $(\varphi', i)$
        if $j > 0$ then
            $i \leftarrow j$
            $V_i \leftarrow \varphi'_i$
            $V' \leftarrow V' \cup \{V_i\}$
        else return $\varphi^\sigma$
    $V'' \leftarrow \{v \in (V \backslash V') | (v, u) \in E, u \in V'\}$
    $\varphi'' \leftarrow \{l \in (\varphi \backslash \varphi')\}$
    assignment $(V'', \varphi'')$
    $c \leftarrow B_c(G, \varphi')$
    if $c > b$ then
        $j \leftarrow$ backtracking $(\varphi', i)$
        if $j > 0$ then
            $i \leftarrow j$
            $V_i \leftarrow \varphi'_i$
            $V' \leftarrow V' \cup \{V_i\}$
        else return $\varphi^\sigma$
    if $i = n$ then
        $\varphi^\sigma \leftarrow \varphi'$
        $b \leftarrow b - 1$
        $j \leftarrow$ backtracking $(\varphi', i)$
        if $j > 0$ then
            $i \leftarrow j$
            $V_i \leftarrow \varphi'_i$
            $V' \leftarrow V' \cup \{V_i\}$
        else return $\varphi^\sigma$
    $i \leftarrow i + 1$
```

V_7 and V_8 could be labeled without exceeding the current bound b in further iterations. However, $B_c(G, \varphi'_k) \leq b$ could not be guaranteed for V_9. In this case, the evaluation of φ_k is stopped making a backtracking which produces the next possible labeling φ_{k+1} in lexicographic order.

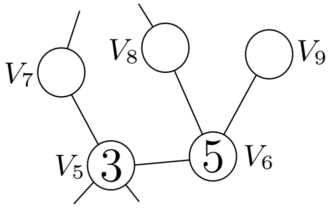


Fig. 2. Example of a partially labeled graph

Table 1. Table assignment vertex-label

	φ''_k			
V''_k	*1*	*4*	*9*	*10*
v_7	**2**	1	5	3
v_8	4	**1**	4	5
v_9	4	1	4	5

After preliminary experimentation with this *basic assignment strategy* we have observed that it could be improved. The amelioration consists in also analyzing the assignments of the adjacencies between the vertices belonging to the set V''_k. Although it represents an additional computational effort, it permits to find at this step vertices producing a cost which exceeds the current bound b. The main advantage of this *improved assignment strategy* is that it is able in certain cases to stop the evaluation of φ_k earlier than the basic assignment strategy, which reduces importantly the size of the search space.

3.3 Backtracking Strategy

A backtracking occurs if the cyclic bandwidth of the partial labeling φ'_k exceeds the current bound b when its i-th label is added. This backtracking procedure is performed as described in Algorithm 2. First, the best element in the set φ''_k of available labels is identified. It is the first label $l \in \varphi''_k$ which guarantees that the resulting cyclic bandwidth $B_c(G, \varphi'_k)$ is lower than the current bound b when this label is assigned to the i-th position of the partial labeling φ'_k. If there is no such label meeting this condition, the $(i-1)$-th position of φ'_k is analyzed (a backtracking occurs), the set φ''_k is updated and its best label is searched again. This process is repeated until the best label is found or the complete set φ'_k was analyzed.

Algorithm 2. Backtracking strategy

```
backtracking ($\varphi'$, $i$)
$\varphi''$ Available labels
while true do
    $\varphi'_i \leftarrow$ best($\varphi''$)
    if $\varphi'_i = \emptyset$ then
        $i \leftarrow i - 1$
        if $i > 0$ then
            $\varphi'' \leftarrow \varphi'' \cup \{\varphi'_i\}$
            $\varphi'_i \leftarrow \emptyset$
        else return $i$
    else return $i$
```

4 Experimental Setup

In order to assess the performance of the proposed B&B algorithm introduced in Sect. 3 it was was coded in C and compiled with *gcc* using the optimization flag $-O3$. It was run sequentially into a cluster composed of 4 processors Xeon X5650 equipped with six cores at 2.66 GHz, 32 GB of RAM and Linux operating system.

Due to the deterministic nature of the algorithm, it was executed only one time over each of the selected benchmark instances. The maximum CPU time allowed for each execution was predefined to 36 hours.

4.1 Compared Algorithms

For the computational experiments two different versions of the B&B algorithm described in Sect. 3 were implemented. The main difference between these two versions is the assignment strategy employed. The first one, denoted B&B1, employs the *basic assignment strategy*, which employs the Hungarian method to solve the problem of finding the cost (cyclic bandwidth) of assigning each label $l \in \varphi''_k$ to each possible vertex $v \in V''_k$. The second version, called B&B2, uses the *improved assignment strategy* which also analyzes the assignments of the adjacencies between the vertices belonging to the set V''_k (see Sect. 3.2).

4.2 Benchmark Instances and Performance Assessment

As it was mentioned at the end of Sect. 2, there has been no previous research investigating neither exact nor approximate algorithms for solving the CB problem in the case of general graphs. There exists no standard test-suite reported in the literature for evaluating the performance of this kind of algorithms. For this reason a test-suite composed of 20 benchmark instances is proposed in this paper.

Table 2. Characteristics of the test-suite used in the experiments. It consists of 20 instances representing 4 different classes of graphs.

Graph	$\lvert V\rvert = n$	$\lvert E\rvert = m$	$ED = 2m/n(n-1)$	$\mathcal{T}$
cycle20	20	20	0.105	1
cycle25	25	25	0.083	1
cycle30	30	30	0.069	1
cycle35	35	35	0.059	1
cycle40	40	40	0.051	1
grid5x4	20	31	0.163	4
grid5x5	25	40	0.133	5
grid5x6	30	49	0.113	5
grid5x7	35	58	0.097	5
grid5x8	40	67	0.086	5
path20	20	19	0.100	1
path25	25	24	0.080	1
path30	30	29	0.067	1
path35	35	34	0.057	1
path40	40	39	0.050	1
tree21	21	20	0.095	3
tree25	25	24	0.080	4
tree31	31	30	0.065	4
tree33	33	32	0.061	4
tree35	35	34	0.057	4

The test-suite comprises 4 different classes of standard graphs: cycles, grids, paths and perfect binary trees. All these instances have a number of vertices between 20 and 40 and are publicly available[1]. The characteristics of these instances are detailed in Table 2, where the first three columns indicate the name of the graph as well as its number of vertices and edges. The edge density $(2|E|/|V|(|V|-1))$ of each graph is depicted in Column 4, while the last column presents the theoretical lower bounds $\mathcal{T}$ reported in [10,13,12].

The criteria used for evaluating the performance of the compared B&B algorithms are the same as those used in the literature [14,15]: the total number of partial solutions (φ'_k) examined during the search process, the total number of backtracking operations occurred and the expended CPU time in seconds. For these criteria smaller values are better.

5 Computational Results

The experimental comparison between the two versions of the *Branch and Bound* algorithm introduced in this paper (B&B1 and B&B2) was performed over the

[1] http://www.tamps.cinvestav.mx/~ertello/cbmp.php

benchmark instances described above and employing the same computational platform introduced in Sect. 4.

The results from this experiment are summarized in Table 3, which lists in the first column the name of the benchmark instance. For each compared algorithm this table presents: the total number of partial solutions φ'_k evaluated during the search process ($\#\varphi'_k$), the total number of backtracking operations occurred (B), and the CPU time in seconds (*time*) expended for finding the optimal solution $\varphi*$. Instances marked with the symbol "—" indicate that they could not be solved within the maximum allowed time of 36 hours. Given that we are comparing exact algorithms, it is not necessary to report the cyclic bandwidth achieved as it is always the optimum value.

Table 3. Performance comparison between two different versions (B&B1 and B&B2) of the *Branch and Bound* algorithm proposed in Sect. 3

	B&B1			B&B2		
Graph	$\#\varphi'$	B	*time*	$\#\varphi'$	B	*time*
cycle20	2331	2042	0.01	586	487	0.01
cycle25	12900	11867	0.12	627	514	0.01
cycle30	88516701	81990664	1007.90	2223754	2089272	43.10
cycle35	465326343	431788129	4184.05	46183164	43443834	449.58
cycle40	397409592	372614337	4996.08	23409592	22614337	531.08
grid5x4	123230	113472	0.82	18280	17002	0.16
grid5x5	2674635	2517186	29.32	562911	532316	8.43
grid5x6	144312826	135950146	3016.46	3253192	3099321	84.63
grid5x7	—	—	—	18490735	17431435	426.32
grid5x8	—	—	—	160963339	155466427	14144.13
path20	695	585	0.01	559	457	0.01
path25	19997	18023	0.07	2498	2227	0.01
path30	101821	93156	7.63	2847814	2513041	0.60
path35	74072	70352	0.83	10393	9815	0.13
path40	—	—	—	10045277883	9116835089	34023.25
tree21	147232	133651	0.47	61483	56644	0.23
tree25	223355577	204264652	763.27	28450274	26320726	110.92
tree31	707968183	663767447	3552.96	220801934	208577566	1240.82
tree33	2248823425	2121853157	10965.57	882035992	836064022	4588.00
tree35	—	—	—	16419099889	15388330578	11288.81

Analyzing the data presented in Table 3 lead us to the following main observations. First, the most time-consuming algorithm is B&B1, despite B&B2 employs the improved assignment strategy which demands an additional computational effort. This is possible because the improved assignment strategy used by B&B2 is able to importantly reduce the size of the explored search space (see column $\#\varphi'_k$).

Second, one observes that B&B1 was not able to find a solution within the time limit of 36 hours for the following classes of graphs with $n \geq 35$: grids, paths and perfect binary trees. On the contrary, B&B2 found an exact solution for all the selected instances employing at most 9.45 hours and consistently a lower number of backtrack operations (B) than B&B1.

The comparison of the computational time expended by B&B1 and B&B2 is better illustrated in Fig. 3. The plot represents the studied instances (ordinate) against the CPU time in seconds expended by the two compared algorithms using a $\log_{10}$ scale (abscissa). From this graph one clearly observes that B&B2 consistently expends much less CPU time than B&B1.

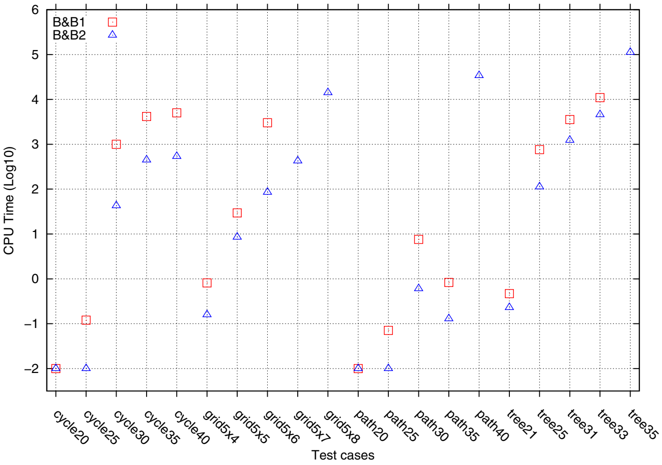


Fig. 3. Consumed CPU time comparison between two different versions (B&B1 and B&B2) of the proposed *Branch and Bound* algorithm

6 Conclusions and Further Work

In this paper a new *Branch and Bound* (B&B) algorithm for the Cyclic Bandwidth (CB) problem was introduced. It is based on a lowest-cost search strategy which explores the branches having the lowest cost while it cuts the branches with higher cyclic bandwidth.

Two different versions of this B&B algorithm, employing slightly different assignment strategies, were implemented. The main difference between these two versions is that the first one, denoted B&B1, employs a *basic assignment strategy* based on the Hungarian method [8] for efficiently finding the cost of assigning every available label in a partial solution to each possible non-labeled vertex; while the second, called B&B2, uses an *improved assignment strategy* which additionally analyzes the assignments of the vertices belonging to the set of unlabeled vertices adjacent to those in the set of already labeled vertices.

Given the lack in the literature of a standard test-suite for evaluating the performance of algorithms for solving the CB problem, a new benchmark composed of 20 instances, comprising 4 different classes of standard graphs (cycles, grids, paths and perfect binary trees), was proposed in this work.

A performance experimental comparison between the two versions of the *Branch and Bound* algorithm introduced in this paper (B&B1 and B&B2) was performed over the benchmark instances proposed. The results from this comparison show that B&B2 consumes much less CPU time (up to 97% in some cases), because it analyzes fewer partial solutions before finding the lower bound of the instances. This gives us the opportunity to find exact solutions for graphs than B&B1 was unable to solve.

This work opens up a range of possibilities for future research. Currently, we are interested in hybridizing our B&B algorithm with some kind of metaheuristic in order to speed the convergence time of the resulting algorithm, inspired in the ideas presented in [15]. The parallelization of the B&B algorithm introduced in this paper also represents an interesting issue for future work.

Acknowledgments. This research work was partially funded by the following projects: CONACyT 99276, Algoritmos para la Canonizacin de Covering Arrays; 51623 Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas.

References

1. Leung, J., Vornberger, O., Witthoff, J.: On some variants of the bandwidth minimization problem. SIAM Journal on Computing 13(3), 650–667 (1984)
2. Bhatt, S.N., Thomson Leighton, F.: A framework for solving VLSI graph layout problems. Journal of Computer and System Sciences 28(2), 300–343 (1984)
3. Rosenberg, A.L., Snyder, L.: Bounds on the costs of data encodings. Theory of Computing Systems 12(1), 9–39 (1978)
4. Chung, F.R.K.: Labelings of graphs. In: Beineke, L.W., Wilson, R.J. (eds.) Selected Topics in Graph Theory, vol. 3, pp. 151–168. Academic Press (1988)
5. Hromkovič, J., Müller, V., Sýkora, O., Vrťo, I.: On Embedding Interconnection Networks into Rings of Processors. In: Etiemble, D., Syre, J.-C. (eds.) PARLE 1992. LNCS, vol. 605, pp. 51–62. Springer, Heidelberg (1992)
6. Papadimitriou, C.H.: The NP-completeness of the bandwidth minimization problem. Computing 16(3), 263–270 (1976)
7. Harper, L.: Optimal assignment of numbers to vertices. Journal of SIAM 12(1), 131–135 (1964)
8. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2(1), 83–97 (1955)
9. Zhou, S.: Bounding the bandwidths for graphs. Theoretical Computer Science 249(2), 357–368 (2002)
10. de Klerk, E., Eisenberg-Nagy, M., Sotirov, R.: On semidefinite programming bounds for graph bandwidth. Technical report, Centrum Wiskunde & Informatica (2011)
11. Yuan, J., Zhou, S.: Optimal labelling of unit interval graphs. Applied Mathematics, A Journal of Chinese Universities 10B(3), 337–344 (1995)

12. Lam, P.C.B., Shiu, W.C., Chan, W.H.: Characterization of graphs with equal bandwidth and cyclic bandwidth. Discrete Mathematics 242(3), 283–289 (2002)
13. Lam, P.C.B., Shiu, W.C., Chan, W.H.: On bandwidth and cyclic bandwidth of graphs. Ars Combinatoria 47(3), 147–152 (1997)
14. Martí, R., Campos, V., Piñana, E.: A branch and bound algorithm for the matrix bandwidth minimization. European Journal of Operational Research 186(2), 513–528 (2008)
15. Palubeckis, G., Rubliauskas, D.: A branch-and-bound algorithm for the minimum cut linear arrangement problem. Journal of Combinatorial Optimization, 1–24 (2011), doi:10.1007/s10878–011–9406–2

The Evolution of Cooperation in File Sharing P2P Systems: First Steps

María Esther Sosa-Rodríguez and Elizabeth Pérez-Cortés

Universidad Autónoma Metropolitana Unidad Iztapalapa
Electrical Engineering Department

Abstract. In order to cope with the free-riding problem in file sharing P2P systems, two kinds of incentive mechanisms have been proposed: reciprocity based and currency based. The main goal of this work was to study the impact of those incentive mechanisms in the emergence of cooperation in file sharing P2P systems. For each kind of incentive mechanism we designed a game and the outcome of this game was used as a fitness function to carry out an evolutionary process. We were able to observe that the Currency Game obtains an enough cooperative population slightly faster than the Reciprocity Game but, in the long run, the Reciprocity Game outperforms the Currency Game because the final populations under the former are consistently more cooperative than the final populations produced by the latter.

Keywords: P2P systems, free-riding, incentive mechanisms, evolutionary algorithm.

1 Introduction

Peer to peer systems (P2P) are composed by autonomous nodes that organize themselves in order to share resources. In P2P systems, each node is a potential server and a client and this characteristic makes them naturally scalable and fault tolerant [1][2]: more peers means more resources and a higher service capacity meanwhile, if a peer leaves the system or fails, another one is able to provide the service. The P2P model is only challenged by peers autonomy that entitles them to decide when to join and leave the system and also what resources they share during their participation. In other terms, scalability and fault tolerance of P2P applications largely depend on the level of cooperation of autonomous nodes.

We place our research in the context of a file sharing P2P (FSP2P) system where files are uploaded by owners and cooperative peers store copies in order to be, later, able to serve download requests. The value of a FSP2P is associated with the amount of shared content[1]. The more content is in the system, the better it is for the clients.

As FSP2P applications became popular, peers having rational behavior appeared. Those peers are named *free-riders* because they attempt to download

[1] In this work, we use indistinctly the terms content and file as well as node and peer.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 151–162, 2013.

files at the lowest cost by deviating from standard protocols [3]. Free-riders pretty soon drove the attention of the community to the design of incentive mechanisms to avoid them. An incentive mechanism mainly encourages nodes to cooperate, detects free-riders and prevents them to get services.

Roughly speaking, there are two types of incentive mechanisms consistently identified in the literature [3][4]: reciprocity based and currency based. In reciprocity based incentive mechanisms, peers give to receive. Each peer has information about the behavior of other peers and uses this information to decide whether to provide or not a service to them. Reciprocity can be direct or indirect. It is direct if peers only consider each other behavior during the current interaction, as in the case of BitTorrent [5]. In the indirect case, information on the past behavior of each peer is collected and used in the evaluation. In currency based incentive mechanisms, peers pay for the service and are paid for their contributions with a currency that can be virtual and/or real.

The main goal of this work was to study, under an evolutionary approach, the impact of incentive mechanisms in a FSP2P system. We modeled the interaction of nodes as a game [6] where nodes follow predefined strategies, issue download requests and an incentive mechanism is applied. The outcome of this game is the successful download ratio and this measure is used as fitness function to carry out an evolutionary process. We studied the impact of each incentive mechanism in the degree of cooperation in the system from two aspects, the way it augments the collective capacity of the system to store all the files and how it modifies the initial composition of the population.

The rest of this article is structured as follows, in section 2 we describe the previous efforts to model P2P systems using games. In section 3, we present the Reciprocity Game and the Currency Game that model the FSP2P system for the two classes of incentive mechanisms. The evolutionary process is presented in section 4 and the experiments we carried out and their results are described and discussed in section 5. Finally, conclusions are outlined in section 6.

2 Related Work

In a FSP2P system, the main decision a peer faces, to store a copy of a file or not, is a strategical situation. Peers have to chose between their individual interests and those of the community: if a peer stores a copy to make it available to the others, it pays some costs but contributes to the value of the system, if it does not, it has no costs but the system may lose value. This situation can be modeled as a game as defined in Game Theory [6]: peers are the players, that have two possible actions and obtain a payoff that depends on the choices made by other peers and the own choice.

During the last decade, several research efforts [7][8][9] have used the Game Theory framework of knowledge to model the strategical situation that peers face. This approach considers that peers are all rational players that always seek the action that provides the best outcome for themselves. The goal of this approach is to find a set of actions where each peer plays the best possible response, i.e. a Nash equilibrium, and use this result to predict the behavior of the system.

Although several assessment studies have confirmed the presence of rational peers in P2P systems, the existence of non rational behaviors, as like altruism has also been observed [10][11]. This reality can be better modeled using Evolutionary Game Theory (EGT) as proposed in [12][13][14]. EGT eliminates the hypothesis of rationality and includes the dynamic aspect of P2P systems. In these works, peers are players following an encoded strategy whose survival depends, on the one hand, on the utility obtained by the strategies when interacting with each other and, on the other hand, on the proportion of the population that follows each strategy. EGT allows us to predict the prevalence of strategies throughout time and the stability of the configuration of a population.

In our case, we are interested in the emergence of cooperative behavior as a result of the application of an incentive mechanism; we do not propose specific strategies but we do want to study the resulting ones. In consequence, we work with a EGT-like approach in the sense that we propose a game for each incentive mechanism where rationality is not assumed but players follow an inherited strategy, and then, the utility that each peer obtains is used as a fitness function for an evolutionary process.

3 The Cooperation Games

In this section, after defining some important concepts and the elements of every game, we present a Generic Cooperation Game (GCG) that models the FSP2P system independently of the incentive mechanism. Then, we propose a game for each class of incentive mechanisms, called Reciprocity Game and Currency Game. They are both based in the GCG and their differences, born from the incentive mechanism they use, are presented in the subsections 3.4 and 3.5 respectively.

3.1 Preliminary Concepts

Availability of peers and files are central concepts in FSP2P systems. On the one hand, the *availability of a peer* models its transient character and it is a number that tell us how likely is to find the peer on the FSP2P system in a given instant. This number depends on the owner and it has been shown that its value follows a Weibull distribution [15]. On the other hand, the *availability of a file* is also a probability, the one of finding the file in a given instant and, in this case, it is determined by the replication technique and the availability of the peers that participate in the storage. In this first work, we considered full replication, that is, when a peer decides to cooperate, it stores and distributes a full copy of the file. In this case, the availability of the file f^i is given by the next expression:

$$f^i_{\text{Availability}} = 1 - \prod_{j=1}^{k} (1 - p^j_{\text{Availability}}) \qquad (1)$$

where k is the number of current copies and $p^j_{\text{Availability}}$ is the availability of the peer that stores the copy number j.

We consider that each shared file has availability requirements and if the system is not able to provide them, the file is lost for practical uses. Also we have to consider that the actual amount of copies of a file that a FSP2P system stores depends, not only on the amount of peers willing to store and share it, but also on other factors like performance, cost, etc. In other words, meanwhile a peer can chose to store a file and share a copy or not, this does not mean that the file is stored in that peer or even stored in the system. Therefore, an *storage event* is recalled by a peer as a triplet: $\langle Coopperate, SystemAvailable, PeerAvailable\rangle$ where *Cooperate* is T if the peer determined to store and share a copy of the file and F otherwise, *SystemAvailable* is T when the file is not lost and F when it is and finally, *PeerAvailable* is T when the peer stores the file and F otherwise. This last entry is meaningful only when the second one is T. Lets notice that an storage event can be represented by a three bit string. A peer is able to keep track of the recent history in a *memory* and might use this information to select its next action.

Clearly, when a peer participates in a FSP2P system, it intends to download content. Therefore, it will issue download requests and some of them will succeed meanwhile others will not. The later situation arises when the file was lost or when the FSP2P system decides that this peer has not been cooperative enough to deserve the service. Each peer has a $successful\ download\ ratio$ (sdr) that is calculated as the quotient between the successful downloads and the total amount of issued download requests. The *sdr* value represents how fit is the strategy of the peer for the particular FSP2P system. We said that a peer is more successful than another peer if it has a bigger *sdr* value.

Finally we define the *success* of a FSP2P system as the percentage of files that are stored in the system w.r.t. those that were submitted to it. A FSP2P system is *cooperative enough* when it is able to store the total amount of files 100%. We said that a population X is more cooperative or more successful than another population Z if X is able to keep a bigger percentage of files than Z.

3.2 Players and Strategies

In the GCG, each peer is a player with a *memory* of size m where it tracks down the last m *storage events* and a *strategy* that specifies an action for each possible value of the *memory.* Since a *storage event* is a three bit string (c.f. subsection 3.1), to recall m of them we need $3 * m$ bits.

Formally we define a *strategy* S as a sequence $s_1 \cdots s_{2^{3*m}}$ where $\forall i, s_i \in$ {*share*, *no-share*}. A *share* action indicates that the peer will store and share a copy of a content whereas *no-share* means that it will not. We say that a strategy A is more cooperative than a strategy B if A includes the action *share* more times than B.

Lets notice that the strategy of a *free-rider* is a sequence of *no-share* actions meanwhile the one corresponding to an *altruist* is a sequence of *share* actions. A randomly generated strategy is called a *honest* strategy as, in average, it tends to contain the same amount of *share* and *no-share* actions.

Summarizing, each player has the following attributes:

- *availability*: probability of being in the system at a given instant,
- *requests*: total number of download requested,
- *downloads*: number of successfully downloaded files,
- *sdr*: successful download ratio = $downloads \div requests$,
- *memory*: the binary description of the last m *storage events* and
- *strategy*: sequence of 2^{m*3} actions

Peers have an unlimited storage space so that a cooperative behavior can not be blurred by its storage capacity. We believe that the current size of hard disks allows us to make this assumption without losing generality.

3.3 The Generic Cooperation Game

A GCG is composed by a sequence of N turns. As in [8][9], on each turn an uniquely identified file f^i is submitted to the system to be stored with a $f^i_{\text{RequiredAvailability}}$. A turn has two phases:

1. Storage phase: during this phase, each peer determines its action independently and simultaneously using its *memory* and its *strategy*. Depending on the result of this phase and the incentive mechanism, the file is stored in the system or lost.
2. Download phase: during this phase, each peer randomly decides if it issues or not a request to download a file f^r where $r \in \{1...i-1\}$. If it does, its *requests* counter is augmented by one. It is up to the incentive mechanism to decide whether a download request is granted or not. When the request is granted and the file is stored in the system, the peer increments its *downloads* counter by one.

At the end of the game, after the N turns, we calculate the *sdr* of each peer (the fitness of its strategy) and the *success* of the FSP2P system.

3.4 The Reciprocity Game

In reciprocity based incentive mechanisms, each peer has a *reputation* that is built during the game. In our model, the *reputation* is the proportion of files a peer is willing to store even if it is not elected to do so.

In the Reciprocity Game (RG) a peer will select the action indicated in $strategy[memory]$. Once all the peers have chosen their actions, using the expression (1), we calculate the potential availability of the file $f^i_{\text{PotentialAvailability}}$ considering that we store a copy on each cooperative peer, then we compare this number against the $f^i_{\text{RequiredAvailability}}$. If

1. $f^{i}_{\text{RequiredAvailability}} = f^{i}_{\text{PotentialAvailability}}$, the file is stored on the system.
2. $f^{i}_{\text{RequiredAvailability}} < f^{i}_{\text{PotentialAvailability}}$ a subset of the cooperative peers, large enough to ensure the requested availability, are randomly elected to store the file.
3. $f^{i}_{\text{RequiredAvailability}} > f^{i}_{\text{PotentialAvailability}}$ the file is lost.

In the RG a request from the peer will be granted if it has an enough good reputation. The amount of requests that are granted for a peer is in direct proportion to its reputation, a peer having a reputation between $0.2 * i$ and $0.2 * (i + 1)$ included, for $i \in \{0, 1, 2, 3, 4\}$, will be granted up to $20 * (i + 1)\%$ of its download requests.

3.5 The Currency Game

In currency based incentive mechanisms, peers pay for the service and are paid for their contributions with a virtual currency.

In this scenario, the Currency Game (CG), peers have two additional attributes, a *storage price* and a *capital*. The first one represents the amount of currency the owner of a file will be charged to store a copy in the peer. This amount is related to the availability of the peer. The *capital* represents the current amount of currency the peer owns. As for files, they have two additional attributes called *rent* and *download price* that represent respectively, the amount of currency they can pay to be stored and the price they will demand to be downloaded.

A peer will cooperate if the action indicated in $strategy[memory]$ is $share$ and the *rent* of the file is at least equal to the *storage price* of the peer. Cooperative peers will be ordered on the base of its *availability* per *storage price* ratio and then, the best of them will be chosen until the *availability* of the file is reached or the *rent* is exceeded. In the former case the elected peers are paid *storage price*. In the latter case, the file is lost and nobody is paid.

A request of a peer will be granted if it has enough currency to pay the *download price* of the file.

4 Evolving Cooperative Behavior

Up to now we have defined two games, the RG and the CG, that provide a way to evaluate the fitness of strategies in a FSP2P system. This section is devoted to the description of the elements of the evolutionary process that we applied in order to obtain more fitted strategies, that is, strategies that allow peers to maximize the amount of downloaded files under a specific incentive mechanism.

The elements of the evolutionary process presented here are largely inspired in the pioneer work of R. Axelrod [16] on the evolution of cooperation in the Prisoner's Dilemma game. For some of them we tested different techniques[2] but did not obtain better results.

[2] Due to the lack of space these tests are not presented in this paper.

Evolutionary Process

Given a number of generations, a game G, a number of rounds N, a size for the population T and a specific proportion of types of strategies in it (c.f. 3.2), we perform the following steps:

1. Initialization: we build a population of nodes whose strategies are compliant with the given specification. Those strategies are the individuals of the evolutionary process. The initial population becomes the current population.
2. Evaluation: to evaluate individuals, the current population plays the game G during N rounds and, at the end, we calculate the *success* of the population and the *sdr* (successful download ratio) of each node ($downloads \div requests$).
3. Ranking: nodes are separated in three groups depending on its *sdr* value as follows (σ stands for the standard deviation of the *sdr* values):
   i) if $sdr < (average(sdr) - \sigma)$ they go to group 0,
   ii) if $(average(sdr) - \sigma) \leq sdr \leq (average(sdr) + \sigma)$ they go to group 1 and
   iii) if $sdr > (average(sdr) + \sigma)$ they go to group 2.
4. Selection: $T/2$ couples of parent nodes are selected. If a peer is in group 1 it is selected once and if it is in group 2 it is selected twice. Peers in group 0 are discarded.
5. Crossover and Mutation: each couple of parents produces a couple of children nodes. In order to obtain the children, two new nodes are created and their strategies are obtained by the application, with a probability P_c, of one point crossover on the strategies of the parents and then, performing on the resulting strategies an uniform mutation with a probability P_m. The *availability* (and the *storage price* for the CG) of the children nodes are inherited from the more successful parent.
6. The whole set of children becomes the current population and the process is repeated since the Evaluation step.

The evolutionary process stops when the prefixed number of generations is reached.

5 Experiments and Results

In order to study the impact of the incentive mechanisms, we implemented a modular Game Simulator using Java 1.6 and we executed the evolutionary process on predefined populations. For the sake of comparison, besides the CG and the RG, we also programmed a game without any incentive mechanism, named Free Game (FG) where download requests are granted randomly.

5.1 Parameters

All the experiments were carried out for a population of 500 nodes with a *memory* of size 3 and strategies of size 512. The *availability* of each node was obtained using a Weibull distribution with form parameter equal to 1. In the case of the Currency Game, initial *capital* was 0 and the *storage price* was calculated randomly on a range that depends on the *availability* of the node as follows: for a node whose *availability* is in $[1\%, 34\%)$, the range was $[0.01, 0.34)$, if the *availability* is in $[34\%, 67\%)$ then, the *storage price* is in $[0.34, 0.67)$ and if the *availability* is in $[67\%, 100\%]$, the *storage price* is in $[0.67, 1.0]$. In the case of the Reciprocity Game, the initial *reputation* was 0.

Each game was played for 1000 turns for files with a required availability uniformly distributed in the range $[1\%, 100\%]$. In the case of the Currency Game, the *rent* was uniformly distributed in the range $[0.1, 10]$ and the *download price* in $[0.01, 1]$.

Populations underwent evolutionary process during 500 generations using a crossover probability $P_c = 0.25$ and a mutation probability $P_m = 0.001$.

Each experiment was repeated thirty times and the value that is considered in the plots is the average value on the thirty repetitions.

5.2 Impact on a Free-Riders Population

The goal of this first experiment was to observe the impact of incentive mechanisms on a population of free-riders. First, we wanted to observe how fast each mechanism drives the population to a state where it is able to keep the 100% of the submitted files and second, how the fitness, the successful download ratio of the strategies, evolves under each mechanism.

In Figure 1 we plot the amount of stored files in each generation for the different games. We can observe a clear difference of the speed of the evolution of cooperation between the populations using incentive mechanisms (RG and CG) and the one where none is used (FG). For this configuration, CG shows a faster, although slightly, evolution in the success of the population and a better stability.

In Figure 2 we can observe the changes in the average successful download ratio of peers under the evolutionary process for each game. In this case, the best growing rate is the one of RG that grows steadily until it reaches the perfect value; the second best is for FG that stabilizes around 0.5, this result is coherent with the fact that, in this game, requests are granted on an uniformly distributed random base. Finally, we can observe that, for CG, the average *sdr* never grows, in fact, as there is a fixed amount of currency being distributed among the peers in the system, no matter how it is distributed, the average will not change. This raises an important issue for the configuration of currency based incentive mechanisms because, if there is too much competition to obtain the currency, as in our experiment, peers having a poor successful download ratio will tend to leave the FSP2P system.

5.3 Impact on the Composition of the Final Population

The goal of this experiment was to observe how different initial populations are changed by the evolutionary process under the RG and the CG. The considered

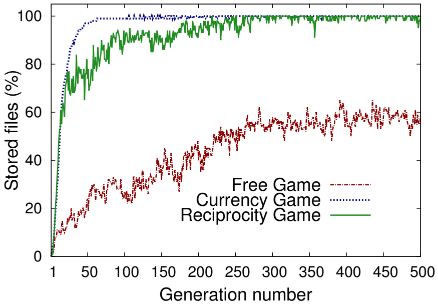


Fig. 1. Success of the population

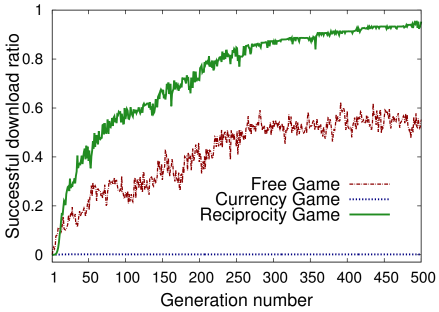


Fig. 2. Successful download ratio

populations, named $P1, P2, P3$ and $P4$ are as follows: $P1$ is a free-rider population, $P2$ is composed by 98% of free-riders and 2% of honest peers, $P3$ is 98% of free-riders and 2% of altruist peers and $P4$ is a population of honest peers. P2 and P3 allowed us to study how a small proportion of honest or altruist peers could boost the emergence of cooperation and drive a population to behave like a honest population (P4).

Programmed Behavior

After 500 generations, we classified the peers in the final population depending on the percentage of *share* actions included in its strategy. We made five groups, each one corresponding to a 20% partition. The results are shown in Figures 3 and 4 for the CG and the RG respectively.

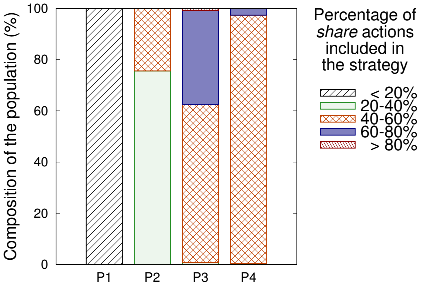


Fig. 3. Programmed behavior CG

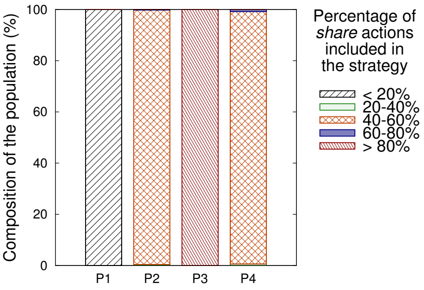


Fig. 4. Programmed behavior RG

For P1, the final population was totally composed of strategies with at most 20% of *share* actions under both incentive mechanism. For P2, in the final population for CG we observe a majority of peers having strategies with at most

40% of *share* actions meanwhile, in the case of RG, it is totally composed by peers with strategies with a percentage of *share* actions between 40% and 60%. For P3, under CG the final population is mainly composed of strategies having more than 40% and less than 80% *share* actions whereas, under RG, it is fully composed of strategies with more than 80% of *share* actions. Finally, for the honest peer population, P4, we do not observe meaningful differences in the composition of the final population. Under both mechanisms the majority of the peers have strategies with a percentage of *share* actions between 40% and 60%.

Summing up, the strategies of the populations produced by RG are more cooperative, include the action *share* more times, than those produced by CG.

Exhibited Behavior

As explained before, in our models a peer determines its actions using the current value of its memory and its strategy, but, even when its strategy is composed of a certain proportion of *share* and *no-share* actions, the peer can behave, for example, as a free-rider (playing always *no-share*) or as an altruist (play always *share*). In other words, peers can exhibit a behavior that differs strongly from the one that appears in its strategy. In consequence, we decide to observe the exhibited behavior, in order to do so, we classified the peers of the final population depending on the percentage of *share* actions they chose to play in the last game. Again, we have a partition at every 20%. The results are shown in Figures 5 and 6 for the CG and the RG respectively.

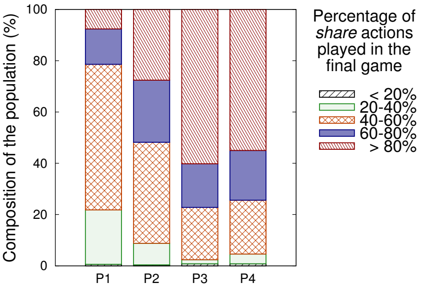


Fig. 5. Exhibited behavior in CG

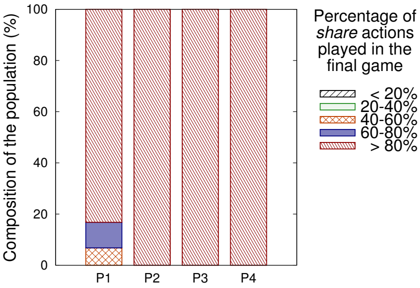


Fig. 6. Exhibited behavior in RG

We can observe that exhibited behaviors include more cooperation, in both RG and CG, than programmed behaviors. On the one hand, under CG, P1 behavior ends up far away from the programmed behavior with more than the 99% playing the action *share* at least 20% of the times and, in the case of P2, P3 and P4, more than 85% chose the action *share* at least 40% of the times. On the other hand, under RG, P1 has only a small fraction of peers that played the action *share* less than 80% whereas in P2, P3 and P4, peers selected the action *share* at least 80% of the times.

Summarizing, independently of the initial composition of the population, the peers of the final populations evolved by RG behave more cooperatively than the peers in the final populations obtained using CG.

Impact of Honesty and Altruism

We can observe, in Figures 4 and 6, that under RG it is enough to introduce 2% of honest peers (P2) to obtain a final population that is close, in its programmed and exhibited behavior, to a honest population (P4). In the case of CG, see Figures 3 and 5, it is the introduction of altruist peers (P3) what produces a composition closer to P4.

6 Conclusions

In this article we studied the evolution of cooperation in a FSP2P system using two different incentive mechanisms. For each one of them we designed a game where peers play inherited strategies and are able to successfully download a portion of the files that they required. Using this measure as a fitness function, we applied an evolutionary process and observed the effect of the incentive mechanisms in the emergence of cooperation, the fitness function and the changes on the composition of the population.

Our results clearly showed that using an incentive mechanism encourage the cooperative behavior. In the case of our proposals, the Currency Game evolves, slightly faster than the Reciprocity Game, an enough cooperative population. However, if the parameters of the system involving currency are not well tuned, this mechanism could lead to the abandon of the FSP2P system due to the steady low successful download ratio of peers. This situation does not arise in the case of the Reciprocity Game.

Also, we were able to observe that, for four specific types of populations, in the quest of cooperative behavior, the Reciprocity Game outperforms the Currency Game. In fact, the final populations under the Reciprocity Game are consistently more cooperative than final populations of the Currency Game. This conclusion is supported by two facts: i) the *share* action appears more times in the strategies of peers in the final populations of the Reciprocity Game than in those of the Currency Game and, ii) in the final Reciprocity Game the action *share* was played a bigger percentage of times than in the final Currency Game, no matter how the initial population was composed.

In the future, we intend to perform a full parametric study on the models of the games and on the evolutionary process, as well as to test different fitness functions. The final goal is to use this approach to study incentive mechanisms used in real systems and take advantage of the results to enhance them.

Acknowledgments. The first author acknowledges support from CONACyT to pursue graduate studies in Science and Technologies of the Information at UAM-Iztapalapa.

References

1. Androutsellis-Theotokis, S., Spinellis, D.: A Survey of Peer-to-Peer Content Distribution Technologies. ACM Computing Surveys **36**(4) (December 2004) 335–371
2. Passarella, A.: Review: A Survey on Content-Centric Technologies for the Current Internet: CDN and P2P Solutions. Computer Communications **35**(1) (January 2012) 1–32
3. Zhang, K., Antonopoulos, N., Mahmood, Z.: A Review of Incentive Mechanism in Peer-to-Peer Systems. In: Proceedings of the First International Conference on Advances in P2P Systems, 2009. AP2PS '09, IEEE (October 2009) 45–50
4. Feldman, M., Chuang, J.: Overcoming Free-riding Behavior in Peer-to-Peer Systems. SIGecom Exchanges **5**(4) (July 2005) 41–50
5. Cohen, B.: Incentives Build Robustness in BitTorrent. In: Proceedings of the First International Workshop on Economics of Peer-to-Peer Systems. (June 2003)
6. Peters, H.: An Introduction to Evolutionary Games. Springer, Berlin Heidelberg (2008)
7. Geels, D., Kubiatowicz, J.: Replica Management Should Be a Game. In: Proceedings of the 10th Workshop on ACM SIGOPS European Workshop. EW 10, ACM (2002) 235–238
8. Chun, B.G., Chaudhuri, K., Wee, H., Barreno, M., Papadimitriou, C.H., Kubiatowicz, J.: Selfish Caching in Distributed Systems: A Game-Theoretic Analysis. In: Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing. PODC '04, ACM (2004) 21–30
9. Khan, S.U., Ahmad, I.: A Pure Nash Equilibrium Guaranteeing Game Theoretical Replica Allocation Method for Reducing Web Access Time. In: Proceedings of the 12th International Conference on Parallel and Distributed Systems - Volume 1. ICPADS '06, IEEE Computer Society (2006) 169–176
10. Kaune, S., Tyson, G., Pussep, K., Mauthe, A., Steinmetz, R.: The Seeder Promotion Problem: Measurements, Analysis and Solution Space. In: Proceedings of 19th International Conference on Computer Communications and Networks 2010. ICCCN '10, IEEE (August 2010) 1–8
11. Hughes, D., Coulson, G., Walkerdine, J.: Free Riding on Gnutella Revisited: The Bell Tolls? IEEE Distributed Systems Online **6** (2005) 1–6
12. Ma, R.T.B., Lee, S.C.M., Lui, J.C.S., Yau, D.K.Y.: Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach. IEEE/ACM Transactions on Networking **14**(5) (October 2006) 978–991
13. Zhang, Q., Xue, H.F., Kou, X.D.: An Evolutionary Game Model of Resources-Sharing Mechanism in P2P Networks. In: Proceedings of the Workshop on Intelligent Information Technology Application 07. IITA '07, IEEE Computer Society (2007) 282–285
14. Wang, Y., Nakao, A., Vasilakos, A.V., Ma, J.: P2P Soft Security: On Evolutionary Dynamics of P2P Incentive Mechanism. Computer Communications **34**(3) (March 2011) 241–249
15. Stutzbach, D., Rejaie, R.: Understanding Churn in Peer-to-Peer Networks. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement. IMC '06, ACM (2006) 189–202
16. Axelrod, R.: Evolving New Strategies: The Evolution of Strategies in the Iterated Prisoner's Dilemma. In: Genetic Algorithms and Simulated Annealing, Pitman (1987) 32–41

Optimal Joint Selection for Skeletal Data from RGB-D Devices Using a Genetic Algorithm

Pau Climent-Pérez[1], Alexandros Andre Chaaraoui[1],
Jose Ramón Padilla-López[1], and Francisco Flórez-Revuelta[2]

[1] Department of Computer Technology,
University of Alicante, Alicante, Spain
{pcliment,alexandros,jpadilla}@dtic.ua.es
http://web.ua.es/dai

[2] Faculty of Science, Engineering and Computing, Kingston University,
Kingston upon Thames, United Kingdom
F.Florez@kingston.ac.uk

Abstract. The growth in interest in RGB-D devices (e.g. *Microsoft Kinect* or *ASUS Xtion Pro*) is based on their low price, as well as the wide range of possible applications. These devices can provide skeletal data consisting of 3D position, as well as orientation data, which can be further used for pose or action recognition. Data for 15 or 20 joints can be retrieved, depending on the libraries used. Recently, many datasets have been made available which allow the comparison of different action recognition approaches for diverse applications (e.g. gaming, Ambient-Assisted Living, etc.). In this work, a genetic algorithm is used to determine the contribution of each of the skeleton's joints to the accuracy of an action recognition algorithm, thus using or ignoring the data from each joint depending on its relevance. The proposed method has been validated using a k-means-based action recognition approach and using the *MSR-Action3D* dataset for test. Results show the presented algorithm is able to improve the recognition rates while reducing the feature size.

Keywords: feature selection, genetic algorithm, RGB-D devices, human action recognition.

1 Introduction

In recent times, a great interest has been put on affordable devices that can capture depth quite reliably. These provide a depth image (D), along with an RGB image (thus RGB-D). The depth image can be further processed to obtain marker-less body pose estimation by means of a skeleton model consisting of a series of joint data, which in turn can be fed to a machine learning algorithm in order to learn and recognise poses, actions, or complex activities.

It is in this field, where it might be interesting to determine which joints have a greater contribution to the success of the learning/recognition method used. Furthermore, it might be also interesting to analyse if these "key joints"

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 163–174, 2013.

are different depending on the dataset or application (e.g. a subset of joints is important for gaming applications, but not for the recognition of Activities of Daily Living –ADLs–).

In this paper, the use of a genetic algorithm for the intelligent selection of joints is presented, which, by using a machine learning algorithm with a cross-validation performed over a well-known dataset, is able to select the most valuable subset of joints to use in order to increase the accuracy of the recognition method.

The remainder of this paper is organised as follows: different methods are presented, this includes the genetic algorithm (Sec. 2), as well as, the action recognition technique (Sec. 3) used; after that, in Sections 4 and 5, experimentation and results are presented; finally, some discussion and conclusions are drawn.

1.1 Body Joint Data Acquisition from RGB-D Devices

The Moving Light Display (MLD) experiments [1] demonstrated the human ability to distinguish different activities from lights attached to the joints of an actor (as stated in [2]). It does not seem risky to hypothesise that joint data contains very valuable information for the recognition of activities by computers; and in addition, that methods that rely solely on the spatial distribution of joints can obtain good performance.

Recently, a number of different applications for human action recognition based on RGB-D devices have been developed. In parallel several datasets have been recorded and publicly offered to be used by researchers. While some are based on gesture/action recognition for Natural User Interfaces (the so-called NUI's) or gaming [3], others try to recognise more complex activities for different applications even using data from the subject–object interaction [4,5,6,7].

Two skeleton models are widely available to work with RGB-D devices. One is the skeleton model used by *Microsoft Kinect SDK*, that has 20 joints available (see Fig. 1). The other is the *OpenNI/NITE* skeleton, which tracks 15 joints.

1.2 Genetic Algorithms for *Optimal* Feature Selection

The *feature subset problem* arises when using feature vectors, since these might contain irrelevant or redundant information that could negatively affect the accuracy of a classifier [8]. In fact, irrelevant and redundant features interfere with useful ones, and most classifiers are unable to discriminate them [9,10].

Otherwise defined, not all the features from the input space are really useful. Feature selection consists in finding a subset of variables that optimises the classification, leading to a higher performance. Also, identifying relevant features is of great importance for classification tasks, as it improves accuracy and reduces computational costs involved [11,12].

In the particular case of skeletal data from RGB-D devices, some points are more characteristic to represent the pose or movement than others [13]. In fact, the joints in the torso (shoulders, spine, torso, waist, hips, etc.) rarely exhibit

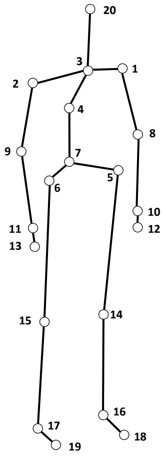


Fig. 1. The 20 joints from a skeleton in the *MSR-Action3D* dataset, numbered using the same order employed in the dataset's X-Y-Z coordinate files

strong independent motion; thus, dimensionality reduction, which improves classification performance, can be applied taking these constraints into account, while retaining the character of the motion [13].

Two main models are presented to do this [8,11]: the *wrapper* model [14], and the *filter* model. The latter selects the features based on *a priori* decisions on feature relevance according to some measures (information, distance, dependence or consistency) [15], but it ignores the *induction* algorithm underneath (that is, the actual algorithm that *induces* a model that classifies the objects into a set of labels or classes). In the former and on the contrary, the feature selection algorithm exists as a *wrapper* around the *induction* algorithm; in the search of a feature subset, the *inductor* itself is used as part of the evaluation function [14].

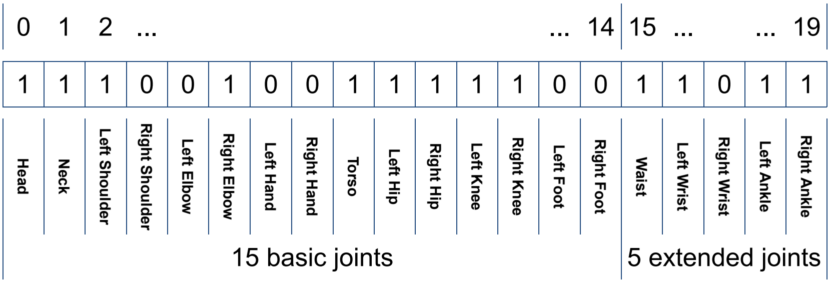
The main disadvantage of the wrapper approach is the time needed for the evaluation of each feature subset [9]. On the other hand, filter-based approaches, although faster, find worse subsets [8].

Genetic algorithms can be applied as wrapper methods, and they are in fact commonly used for the feature selection problem as they have proved appropriate; even so, other methods such as *Tabu Search*, *GRASP*, and other methods perform better [11].

2 A Genetic Algorithm for Skeletal Feature Selection

The main contribution of this paper is the use of a genetic algorithm for the selection of joint features. An individual is coded as a boolean array $\mathbf{u}$ whose elements $\mathbf{u}_j$ are a value that represent the inclusion or exclusion of the jth joint from the data which is passed in to the recognition system.

It is worth noting that, the order of elements in $\mathbf{u}$ has been chosen so that the different skeleton models could be employed. Since one presents only 15 joints, as



0	1	2	...										...	14	15	...		...	19
1	1	1	0	0	1	0	0	1	1	1	1	1	0	0	1	1	0	1	1
Head	Neck	Left Shoulder	Right Shoulder	Left Elbow	Right Elbow	Left Hand	Right Hand	Torso	Left Hip	Right Hip	Left Knee	Right Knee	Left Foot	Right Foot	Waist	Left Wrist	Right Wrist	Left Ankle	Right Ankle
15 basic joints															5 extended joints				

Fig. 2. An array **u**, as the ones used in the genetic algorithm for joint selection. It can be seen that joints 0 to 14 are called *basic joints*, as they are common to both skeleton models presented in Sec. 1.1. On the other hand, joints 15 to 19 are called *extended* for they are only present in the 20-joint model from *Microsoft*.

stated in Sec. 1.1, making **u** model-independent, allows the use of both models (see Fig. 2).

The employed algorithm is a genetic algorithm, with the addition of a taboo list, where unsuccessful individuals are stored to avoid repeating the evaluation, as it is a very time-consuming task.

Concretely, the algorithm performs the following steps:

1. Create a random population. The first individual is always that whose $\mathbf{u}_j = 1, \forall j \in 0..J$ (being J the total number of joints in the selected skeleton model). The rest are randomly created, favouring the inclusion of joints in the feature, by using a probability ($P_1 = 0.75; P_0 = 0.25$, these values have been chosen empirically).
2. Evaluate the fitness for the initial population. To this end, a k-means (see Sec. 3) cross-subject cross-validation is employed (see Sec. 4). Other recognition methods and training sets can be used.
3. Order the population in descending fitness value.
4. Create a new individual by recombination, using a *one-point crossover* selecting the two parents by ranking and the crossover point at random.
5. Mutate the new individual's elements, using a mutation probability $P_m = 0.1$.
6. Assure that:
   - First, the new individual does not exist in the population.
   - Second, the new individual is not included in the *taboo list*.

   If any of these conditions is not fulfilled, then go back to step 4.
7. Add the worse-performing individual to the *taboo list*, and remove it from the population.
8. Return to step 3; repeat until no changes in the best-performing individual are observed during a number of generations max_{gen}.

3 The Action Recognition Method

The objective of this paper is not the development of an action recognition method to be used as a fitness function. Therefore, a state-of-the-art algorithm is used [16,17], which is based on k-means for the creation of a series of *key poses* which are then used during the recognition phase in which a weighting scheme is used to determine the final labelling for the input. The authors in [16] use this scheme for contour-based action recognition, with silhouettes extracted from conventional RGB cameras. The method has been modified to deal with joint data, instead of contour points as used in the original method.

In [16] the *feature* consisted of a series of contour point coordinates (x, y) that were retrieved from silhouettes obtained from background subtraction techniques applied to RGB video streams. These coordinates were extracted in a clockwise fashion: first, the centroid was estimated; then, starting from the leftmost point (relative to the centroid) a determined number of points L were extracted from the silhouette; finally, the coordinates are translated and normalised, to the so-called normalised distance signal $\bar{DS}$.

Following a similar approach and since data from RGB-D devices is used instead, the body pose feature is calculated as follows: first, the 3D *real-world*[1] coordinates for 15 to 20 joints are retrieved from a marker-less body pose estimator based on the depth images obtained; after that, coordinates are translated so that the Torso (marked as '4' in Fig. 1) point is considered the origin $(0, 0, 0)$; then, the distance from each joint to the Torso can be normalised according to the Torso–Neck distance (see Section 4).

This feature is calculated for every skeleton present in the training data. That is, when the training process starts, the skeleton data for each action class are retrieved from the available training data (depicted as $T_0..T_N$ in Fig.3, which summarises the process), the feature is extracted for every frame as just explained. Then, each set of skeleton data from the different action sets are clustered using k-means. At this step, different values for K can be provided; this determines the number of class representatives per action (*key poses*) to be retrieved.

After this initial process, the training data is fed back, and the skeleton in each frame of each of the training sequences is evaluated against the *key poses.* If the nearest key pose *kp* returned is of the same action class as the skeleton retrieved from the sequence, the value hits_{kp} is increased; otherwise, missed_{kp} is. This way, after the process has finished, each key pose *kp* is given a weight w_{kp}, such that:

$$w_{kp} = \frac{\text{hits}_{kp}}{\text{hits}_{kp} + \text{missed}_{kp}} \ . \tag{1}$$

What this means is, that a key pose *kp* that received a high number of hits $(\uparrow w_{kp})$, is more representative than a key pose with a lower weight.

[1] When real-world coordinates are retrieved; these are obtained in millimetres from the sensor.

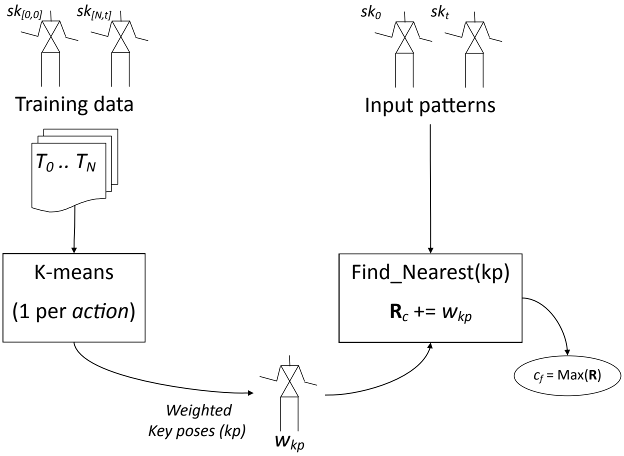


Fig. 3. The work-flow of the proposed action recognition scheme

During recognition, the skeletons from the input patterns are extracted, being sk_t each of the skeletons to test (where t represents the time, as a frame number).

Furthermore, an array of results $\mathbf{R}$ is created. To proceed to recognition, for each of the extracted skeletons sk_t, the nearest available key pose kp is retrieved; then, w_{kp} is added to $\mathbf{R}_c$ being c the action class of kp. The final label $c_f = \max(\mathbf{R})$ is calculated and the corresponding action class c_f (label) is returned.

4 Experimentation

Choosing a Dataset. In this work, the *MSR-Action3D* dataset is used [3]. This dataset contains 20 different actions (see Table 1), performed by 10 different subjects and with up to 3 different repetitions. This makes a total of 567 sequences. Since a dataset is used in the current work, the election of a skeleton model is constrained to the one used in that specific dataset. In this case, it is the 20-joint model described in Sec. 1.1.

Validating the Action Recognition Method. The method presented in Sec. 3 has been validated; to do so, a cross-subject validation is used, in which half the subjects are used for training and the remainder are used for test (and *vice-versa*[2]) as performed by various other works [3,4,18].

Two things are worth mention at this point: one is, that even if the proposed method does not outstrip state-of-the-art methods using the same dataset [3,4,18],

[2] In *MSR-Action3D* this means subjects `s01` to `s05`; and subjects `s06` to `s10`. Note this is different from a Leave-one-actor-out cross-validation.

Table 1. Actions in the *MSR-Action3D* dataset

Label	Action name	Label	Action name	Label	Action name
a01	High arm wave	a08	Draw tick	a15	Side-kick
a02	Horizontal arm wave	a09	Draw circle	a16	Jogging
a03	Hammer	a10	Hand clap	a17	Tennis swing
a04	Hand catch	a11	Two-hand wave	a18	Tennis serve
a05	Forward punch	a12	Side-boxing	a19	Golf swing
a06	High throw	a13	Bend	a20	Pick-up and throw
a07	Draw cross	a14	Forward kick		

it is not the current aim, since this work intends to show how intelligent joint selection might affect accuracy; and the second is, the presented method achieved a recognition rate of 65.7 % (with $K = 8$) on the cross-subject test (using all the joints, that is $\mathbf{u}_j = 1, \forall j \in 0..J, J = 20$), which is used as a base value to test other $\mathbf{u}$'s.

Defining a Fitness Function. A cross-subject validation with the whole dataset using the presented recognition method (Sec. 3) is used as the fitness function for the genetic algorithm. To this end, the training and recognition phases were changed so that an *individual* (that is, an array of joint weights $\mathbf{u}_j,\ j \in 0..J$) is used for the distance functions. This way, recognition rates can be calculated when different $\mathbf{u}$'s are passed in. A low value for K is used, $K = 8$, so, after the clustering, eight different *key poses* (class representatives) are obtained for each of the action classes.

Choosing the Best-Performing Population Size. Different population sizes are tested to compare their performance. The proposed algorithm is launched using populations of 2, 5, 10, and 25 individuals plus 1 offspring individual both with and without the Torso-Neck distance normalisation.

Optimal Feature Selection. By means of the proposed genetic algorithm, an *individual* $\mathbf{u}$ is selected, which optimises the accuracy value of the recognition method in use (i.e. the best individual in the population after the evolution). Results and discussions follow.

5 Results and Discussion

Setting the size of the population to 5 individuals and letting it evolve, valuable results have been retrieved. The best results were retrieved when not using the Torso-Neck normalisation. As figures 4 and 5 show, there is an improvement of 5.4 % after $max_{gen} = 500$ generations without change. Furthermore, differences in the confusion matrices obtained before and after the evolution can be observed.

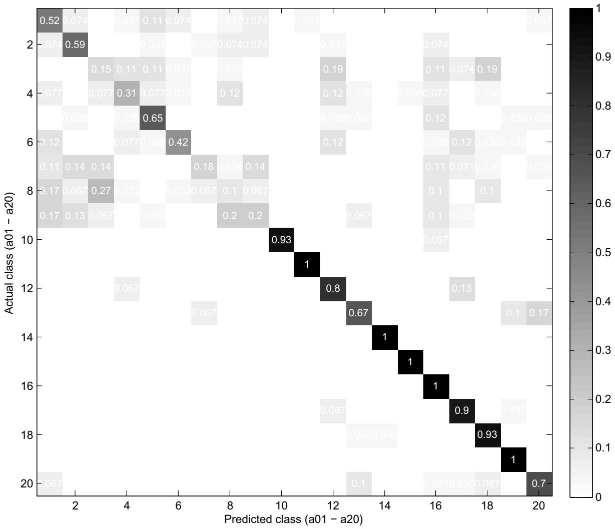


Fig. 4. Initial confusion matrix for generation 0 (individual with all joints selected). The overall accuracy for this generation is 65.7 %. Rows present the actual action classes, namely *a01* to *a20*. Columns present the predicted labels given by the action recognition method in use.

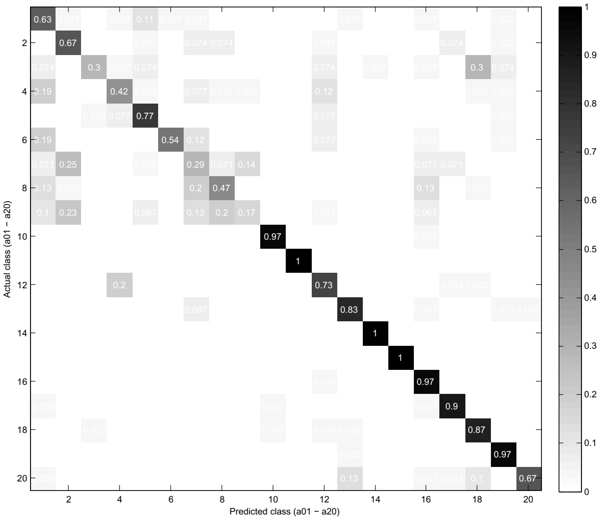


Fig. 5. Confusion matrix after the evolution. The overall accuracy for this generation is 71.1 %. See the previous figure (Fig. 4) for details on interpretation.

The selected skeleton joints are shown in Fig. 6. This shows which joints were considered relevant, and which were rejected since they added noise to the recognition. As an example, it can be observed that the head and the torso points are not included; the head does not present much movement throughout the sequences, and the torso point is used as the centre for normalisation. The feet are also excluded, probably due to the noise added by inference of these joints. The right wrist is included, which might indicate that the right hand is of more value, since a series of arm-related actions (drawing ticks, crosses, circles) are performed only using this arm, as stated by the creators of the dataset [3].

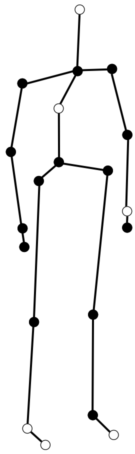


Fig. 6. The 20 joints from a skeleton in the *MSR-Action3D* dataset showing the final feature subset; selected joints are shaded in black, ignored ones are left unshaded

6 Conclusions and Future Work

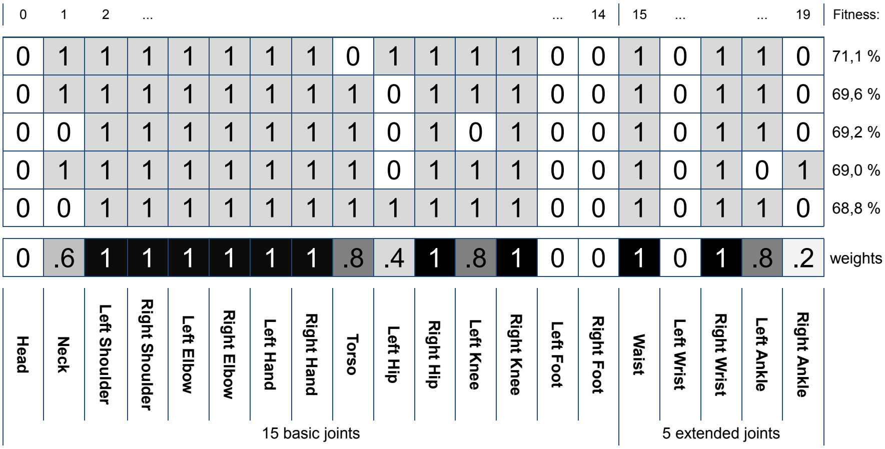
To conclude, the presented algorithm has been able to demonstrate the initial hypothesis that stated that some joints can be considered as ‘noise’ impeding the classifier to achieve better performance. Taking the results into account, it can also be seen that the joint selection algorithm seems to actually take more relevant joint data into account (e.g. the upper right limb); while discarding noisy, inferred or non-moving elements (e.g. head, feet...).

It should be well noted that these results are specific to this dataset and this action recognition method. One interesting future work is to improve the results of other state-of-the-art action recognition methods by means of the genetic algorithm presented.

Figure 7 shows all the individuals in the population once the evolution is completed. Results seem reasonably consistent (e.g. feet are always left out). Furthermore, an average of the joint values for the population in that generation is given at the bottom. Darker spots indicate more relevant joint features; lighter ones, point irrelevant joint information.

Most feature selection approaches reviewed in the literature use binary selection of features; that is, features are either used (`true`, 1) or ignored (`false`, 0). But, it can be also interesting to give a weight to the features that allows better fine-tuning of the relevance of each of them. Perhaps, a feature is very discriminative, and has a greater weight that another one that is not so relevant, but clearly superior than a third one whose contribution is almost null. Thus, when features are given real-valued weights, any weight that moves towards a given maximum indicates that the classification process is sensitive to small changes in that feature [19].

In future works it might be useful to use such real-valued weights for a better, finer tuning of the usefulness of each joint (using averaging of binary-valued arrays –as done in Fig. 7–, or other means for directly producing real-valued arrays). Other future research lines include the use of more complex wrappers, based on Genetic Algorithms (e.g. Memetic Algorithms, that use additional knowledge in the recombination process). It will be also interesting to study how the selected features (joints that are not considered *noise* by a certain algorithm) differ depending on the dataset employed (application-related selection) or the recognition method in use. Finally, other kind of features need to be studied, such as angles instead of distances (angles between the joints, i.e. quaternions or other such means), or other kinds of distance arrays (e.g. *cross distances*, i.e. distances among every pair of joints).



0	1	2	...										...	14	15	...		...	19	Fitness:
0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	1	1	0	71,1 %
0	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	69,6 %
0	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	0	1	1	0	69,2 %
0	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	69,0 %
0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	0	68,8 %
0	.6	1	1	1	1	1	1	.8	.4	1	.8	1	0	0	1	0	1	.8	.2	weights
Head	Neck	Left Shoulder	Right Shoulder	Left Elbow	Right Elbow	Left Hand	Right Hand	Torso	Left Hip	Right Hip	Left Knee	Right Knee	Left Foot	Right Foot	Waist	Left Wrist	Right Wrist	Left Ankle	Right Ankle	
						15 basic joints											5 extended joints			

Fig. 7. The 5 individuals at the end of the evolution. An average of the joint values is given at the bottom. Colour-coding is used to indicate different degrees of relevance or contribution of each particular joint.

Acknowledgements. This work has been partially supported by the Spanish Ministry of Science and Innovation under project "Sistema de visión para la monitorización de la actividad de la vida diaria en el hogar" (TIN2010-20510-C04-02) and by the European Commission under project "caring4U - A study on people activity in private spaces: towards a multisensor network that meets privacy requirements" (PIEF-GA-2010-274649). Alexandros Andre Chaaraoui and Jose Ramón Padilla-López acknowledge financial support by the Conselleria d'Educació, Formació i Ocupació of the Generalitat Valenciana (fellowships ACIF/2011/160 and ACIF/2012/064 respectively). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. Johansson, G.: Visual perception of biological motion and a model for its analysis. Attention, Perception, & Psychophysics 14, 201–211 (1973), doi:10.3758/BF03212378
2. Polana, R., Nelson, A.: Detection and recognition of periodic, nonrigid motion. International Journal of Computer Vision 23, 261–282 (1997)
3. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14 (June 2010)
4. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), Providence, Rhode Island (June 2012)
5. Sung, J., Ponce, C., Selman, B., Saxena, A.: Human activity detection from RGBD images. In: AAAI Workshop on Pattern, Activity and Intent Recognition, PAIR (2011)
6. Ni, B., Wang, G., Moulin, P.: RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1147–1153 (November 2011)
7. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3-d object dataset: Putting the kinect to work. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1168–1174 (November 2011)
8. Cantú-Paz, E.: Feature Subset Selection, Class Separability, and Genetic Algorithms. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 959–970. Springer, Heidelberg (2004)
9. Lanzi, P.: Fast feature selection with genetic algorithms: a filter approach. In: IEEE International Conference on Evolutionary Computation, pp. 537–540 (April 1997)
10. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI 1992, pp. 129–134. AAAI Press (1992)
11. Casado Yusta, S.: Different metaheuristic strategies to solve the feature selection problem. Pattern Recognition Letters 30(5), 525–534 (2009)
12. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. IEEE Intelligent Systems and their Applications 13(2), 44–49 (1998)

13. Raptis, M., Kirovski, D., Hoppe, H.: Real-Time Classification of Dance Gestures from Skeleton Animation. In: Proceedings of the 10th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2011, pp. 147–156 (2011)
14. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proceedings of the 11th International Conference on Machine Learning, pp. 121–129. Morgan Kaufmann, San Francisco (1994)
15. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Boston (1998)
16. Cheema, S., Eweiwi, A., Thurau, C., Bauckhage, C.: Action recognition by learning discriminative key poses. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1302–1309 (November 2011)
17. Chaaraoui, A.A., Climent-Pérez, P., Flórez-Revuelta, F.: An Efficient Approach for Multi-view Human Action Recognition Based on Bag-of-Key-Poses. In: Salah, A.A., Ruiz-del-Solar, J., Meriçli, Ç., Oudeyer, P.-Y. (eds.) HBU 2012. LNCS, vol. 7559, pp. 29–40. Springer, Heidelberg (2012)
18. Yang, X., Tian, Y.: EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor. In: Second International Workshop on Human Activity Understanding from 3D Data in Conjunction with CVPR, 2012, Providence, Rhode Island (2012)
19. Punch, W., Goodman, E., Pei, M., Chia-Shun, L., Hovland, P., Enbody, R.: Further research on feature selection and classification using genetic algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 557–564 (1993)

Dynamic Estimation of Phoneme Confusion Patterns with a Genetic Algorithm to Improve the Performance of Metamodels for Recognition of Disordered Speech

Santiago Omar Caballero Morales and Felipe Trujillo Romero

Technological University of the Mixteca, UTM, Highway to Acatlima, Km. 2.5,
Huajuapan de Leon, Oaxaca, 69000
{scaballero,ftrujillo}@mixteco.utm.mx

Abstract. A field of research in Automatic Speech Recognition (ASR) is the development of assistive technology, particularly for people with speech disabilities. Diverse techniques have been proposed to accomplish accurately this task, among them the use of Metamodels. In this paper we present an approach to improve the performance of Metamodels which consists in using a speaker's phoneme confusion matrix to model the pronunciation patterns of this speaker. In contrast with previous confusion-matrix approaches, where the confusion-matrix is only estimated with fixed settings for language model, here we explore on the response of the ASR for different language model restrictions. A Genetic Algorithm (GA) was applied to further balance the contribution of each confusion-matrix estimation, and thus, to provide more reliable patterns. When incorporating these estimates into the ASR process with the Metamodels, consistent improvement in accuracy was accomplished when tested with speakers of mild to severe dysarthria which is a common speech disorder.

Keywords: Genetic Algorithms, Disordered Speech Recognition, Metamodels.

1 Introduction

Dysarthric speech is different from normal speech as it is affected by breathing and articulation abnormalities which cause performance in Automatic Speech Recognition (ASR) to decrease considerably [9,10,13,15,18]. These abnormalities decrease the speaker's intelligibility and restrict the speaker's phonemic repertoire, thus some sounds or phonemes cannot be uttered or articulated correctly. In ASR this leads to an increase of deletion, insertion, and substitution of phonemes [8,14,15,16].

Most speaker adaptation algorithms are based on the principle that it is possible to apply a set of transformations to the parameters of the acoustic models of an ASR system to move them closer to the voice of an individual [20]. Whilst this

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 175–187, 2013.

has been shown to be successful for normal speakers, it is less successful in cases where phonatory dysfunction is present (and the phoneme uttered is not the one that was intended but is substituted by a different phoneme or phonemes) as often happens in dysarthric speech [7].

In [3] it was proposed that, instead of adapting the acoustic models of the ASR system, the errors made by the speaker at the phonetic level could be modelled to attempt to correct them. This was a concept that was previously proposed in [11] to correct the phoneme output of an ASR system, and in [3] this was further extended to accomplish improvement at the word recognition level. This approach, which made use of a phoneme-confusion matrix to get the estimates of the speaker's pattern of phonetic errors, was also explored in [12], [19], and [17].

In the field of artificial intelligence, a confusion-matrix is a visualization tool typically used in supervised learning. Each column of the matrix represents the instances in a recognised class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (e.g., commonly mislabelling or classifying one as another).

As implementation techniques to incorporate the phoneme confusion-matrix estimates into the ASR process are Weighted Finite-State Transducers (WFSTs) [3,11,17], and Hidden Markov Models (HMMs, Metamodels) [3,12]. However an important issue has remained within these studies, which is the ASR's phoneme output used to estimate the phoneme confusion-matrix.

From the mathematical model of the ASR process:

$$\hat{W} = \max_{W \in L} P(O|W)P(W) \tag{1}$$

the most likely sequence of words $\hat{W}$ given some acoustic observation O can be estimated as the product of two probabilities: $P(W)$, the *prior probability*, which is obtained from the Language Model (L); and $P(O|W)$, the *observation likelihood*, which is obtained from the acoustic model.

$P(W)$ is usually estimated by using N-gram grammars, and $P(O|W)$ is usually modelled by using Hidden Markov Models (HMMs), or Artificial Neural Networks (ANN). For word recognition tasks, the ASR system estimates the sequence of the most likely phoneme models (HMMs) that represent the speech O. Then, this sequence is restricted to form words by incorporating a lexicon. Finally, these words are restricted by the information of the language model $P(W)$ to form valid sequences of words.

The influence of $P(W)$ is very important for the final result, and hence, of the output generated for confusion-matrix estimation. The performance of the implementation techniques (HMMs, WFSTs) [3,12,17] rely critically on the accuracy of the phoneme output sequences used for confusion-matrix estimation [4]. In the studies using the confusion-matrix modelling approach, usually a single confusion-matrix is used for training purposes. In [11] the phoneme sequences were obtained with unrestricted phoneme language. In [4] the training phoneme sequences were obtained from the phonetic transcriptions of the word output of the ASR system. It was found that these sequences were more accurate than those obtained directly at the phonetic level (with a phoneme language model).

However, with this approach, high accuracy in phoneme recognition does not always correlate with high accuracy in word recognition. This is because of the different language models restrictions used for confusion-matrix estimation and word recognition evaluation. Another issue is related to the problem observed when the data available for confusion-matrix estimation is small, which leads to poor estimates, and in practice, this is the normal situation.

In this paper, we propose an extended approach to obtain better estimates of a phoneme confusion-matrix to increase ASR performance for dysarthric speech. Instead of just considering a confusion-matrix estimated with a single language model (either phoneme-based or word-based), we consider a dynamic estimation of diverse confusion-matrices with different language model restrictions. These matrices then are weighted and integrated into a single confusion-matrix, which would have more information about the behaviour of the speaker's confusion patterns across different language model scenarios. The implementation technique for the incorporation of the confusion-matrix into the ASR process is the extended version of the Metamodels (discrete HMMs) [3,12] which was presented in [2]. The weights to balance the contribution of each confusion-matrix are estimated by means of a Genetic Algorithm (GA), which also performs optimization on the structure of the extended Metamodel. When evaluating this approach with a well known database of dysarthric speech, performance was significantly higher when compared with the single confusion-matrix estimation approach, and also, when compared with a Speaker Adaptive (SA) ASR system.

Hence, this paper is structured as follows: in Section 2 information about the integration of the confusion-matrix estimates into the ASR process is reviewed. Then, we present the details of the dynamic estimation of the confusion-matrices. In Section 3 the extended Metamodel for implementation is reviewed, and then in Section 4 the Genetic Algorithm for optimization of the contributions of the dynamic confusion-matrices and the Metamodel is presented. In Section 5 the details of the experiments are presented, which involves the training and evaluation procedures. Finally, in Section 6, the results and future work are discussed.

2 Phoneme Confusion-Matrix as Resource for Error Correction

In Figure 1 an example of a phoneme confusion-matrix is shown, where rows represent the phonemes intended or uttered by the speaker (Stimulus), and the columns represent the decoded phonemes given by the ASR system (Response). The classification of phonemes to estimate a phoneme confusion - matrix is performed by the alignment of two phoneme strings (or sequences):

- P, the reference (correct) phoneme transcription of the sequence of words W uttered by a speaker.
- $\tilde{P}^*$, the sequence of phonemes decoded by the ASR system.

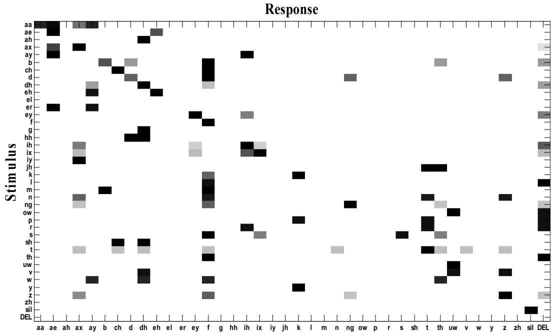


Fig. 1. Example of a phoneme confusion-matrix

As $\tilde{P}^*$ is the system's output, it might contain several errors. Based on the classification performed by the aligner, these are identified as substitution (S), insertion (I), and deletion (D) errors. Thus, the performance of ASR systems is measured based on these errors, and two metrics are widely used for phoneme and word ASR performance: Word Accuracy (WAcc) $= \frac{N-D-S-I}{N}$, and the Word Error Rate(WER) $= 1 - WAcc$. Where N is the number of elements (words or phonemes) in the reference string (P). Thus, the objective of the statistical modelling of the phoneme confusion-matrix is to estimate W from $\tilde{P}^*$. This can be accomplished by the following expression [3]:

$$W^* = \max_P \prod_j^M Pr(p_j)Pr(\tilde{p}_j^*|p_j) \tag{2}$$

where p_j is the j'th phoneme in the postulated phoneme sequence P, and $\tilde{p}_j^*$ the j'th phoneme in the decoded sequence $\tilde{P}^*$ (of length M). Eq. 2 indicates that the most likely word sequence is the sequence that is most likely given the observed phoneme sequence from a speaker. The term $Pr(\tilde{p}_j^*|p_j)$ represents the probability that the phoneme $\tilde{p}_j^*$ is recognized when p_j is uttered, and is obtained from a speaker's confusion-matrix. This element is integrated into the recognition process as presented in Figure 2.

This information then can be modelled by techniques to improve the baseline ASR's output. Evaluation is performed when $\tilde{P}^*$ (which now is obtained from test speech) is decoded by using the "trained" techniques into sequences of words W^*. The correction process is done at the phonetic level, and by incorporating a word-language model a more accurate estimate of W is obtained. In this work, the extended Metamodels are the technique used for the modelling of the confusion-matrix and implementation fo the ASR process. This is presented in Section 3.

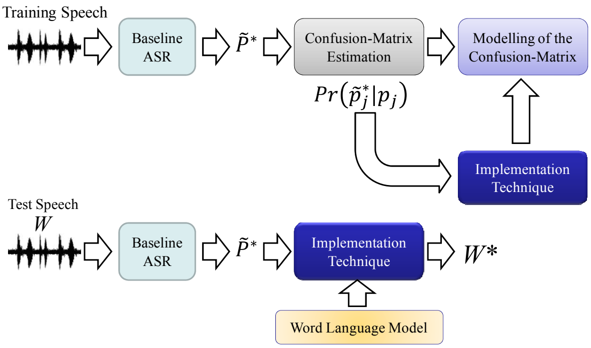


Fig. 2. Training and testing process of the confusion-matrix approach

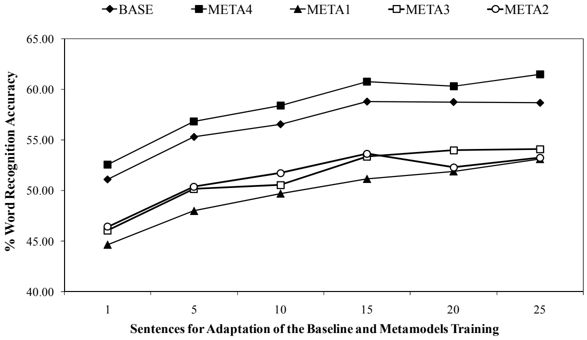


Fig. 3. Comparison of performance of Metamodels trained with different phoneme sequences

2.1 Dynamic Estimation of the Phoneme Confusion-Matrix

As presented in Section 1, the studies that have used the phoneme confusion-matrix as resource for error modelling have only used a single baseline condition to estimate the confusion patterns. This is, a single language model restriction (either phoneme-based or word-based). We consider necessary to study the behaviour of the ASR for different language model restrictions as this affects the phoneme/word output, which is the base for confusion-matrix estimation.

For example, consider Figure 3, which shows the performance of the single phoneme confusion-matrix approach when trained and tested with different number of phoneme sequences generated with different language model restrictions. These results were obtained during a previous exploratory study with the Nemours database of dysarthric speech [1], which is presented in Section 5. The baseline (BASE) represents the adapted word output from a standard ASR system (with continuous HMMs for acoustic modelling). META1 represents the performance of the implementation technique of the Metamodels (see Section

3) trained with the confusion-matrix estimated from phoneme sequences generated with a phoneme-based (PB) language model with no probability restrictions [1](s=0). META2 and META3 are the performance of the same Metamodels with different levels of restrictions for the PB language model (s=10 and s=25). META4 is the performance when the confusion-matrix is estimated from the phoneme transcriptions of the word output generated with a highly restricted word-based (WB) language model (i.e., s >30). As it is presented, there are significant differences in performance.

Also it was observed that there was variability in the patterns of phoneme confusions observed with different levels of language model restrictions. In Figure 4 is shown a sequence of confusion-matrices estimated from phoneme sequence obtained with different grammar scale factors for the language model. In the matrix estimated from phoneme sequences obtained from the transcriptions of words obtained with a WB language model, there are missing confusion patterns that are present in the matrices obtained with a PB language model with different restrictions (s=1,5). These estimates are considered to be important, especially when training data is sparse and estimation of unseen data is required. These confusion patterns potentially can be present in the unseen test set.

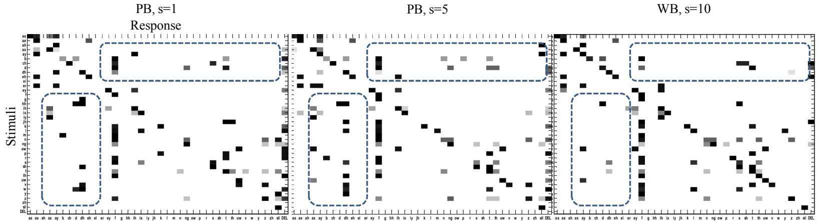


Fig. 4. Comparison of phoneme confusion-matrices obtained with different language model conditions

By considering this information, a more complete confusion-matrix can be estimated. Note however that, incorporating all this information in a single confusion-matrix is not an easy task. This is because some patterns do not contribute in the same way as others. For example, if just a number of matrices are estimated and averaged to get a single matrix, the most dominant patterns across all the matrices will be reinforced in the single matrix. On the other hand, the less dominant, but still important, will be reduced even further. This was considered to be a problem similar to the mixture of gaussian distributions where, in this case, would be a mixture of phoneme confusion-matrices.

We start by taking as reference the definition of the *emission probabilities* of the HMMs, $B = \{b_i(\mathbf{o}_t)\}$, where each term represents the probability of an observation vector $\mathbf{o}_t$ being generated from a state j:

[1] The grammar scale factor is a variable used to set the influence (or restriction) of the language model on the ASR process[20] and is identified as $s \in (0, 30)$.

$$b_j(\mathbf{o}_t) = \sum_{k=1}^{K} C_{jk} N(\mathbf{o}_t, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) \tag{3}$$

In Eq. 3, K denotes the number of mixture-components, C_{jk} is the weight for the k-th mixture component satisfying $\sum_{k=1}^{K} C_{jk} = 1$, and $N(\mathbf{o}_t, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$ denotes a single Gaussian density function with mean vector $\boldsymbol{\mu}_{jk}$ and covariance matrix $\boldsymbol{\Sigma}_{jk}$ for state j.

For this work, K would be the number of confusion-matrices to consider, each one obtained from a different language model condition. C_k the weight which would measure the contribution, or importance, of each k-th confusion-matrix. And $Pr(\tilde{p}_j^*|p_j)^k$ the discrete distribution probabilities associated to the k-th phoneme confusion-matrix (see Eq. 2). Hence, the estimated dynamic phoneme confusion-matrix can be expressed as:

$$Pr(\tilde{p}_j^*|p_j)_{dyn} = \sum_{k=1}^{K} C_k Pr(\tilde{p}_j^*|p_j)^k \tag{4}$$

In Section 4 the algorithm designed to estimate the weights C_k is presented.

3 Extended Metamodels

The Metamodels are a technique proposed to model a speaker's confusion-matrix $Pr(\tilde{p}_j^*|p_j)$ and to incorporate this information into the ASR process [3,12]. These consist of discrete HMMs with the structures shown in Figure 5, which allow the modelling of insertion and deletion patterns. Each state of a metamodel has a discrete probability distribution over the symbols for the set of phonemes (i.e., $Pr(\tilde{p}_j^*|p_j)$), plus an additional symbol labelled DEL (deletion). The central state (C) of a metamodel for a certain phoneme models correct decodings, substitutions and deletions of this phoneme made by the ASR system. States BF and AT model (possibly multiple) insertions before and after the phoneme.

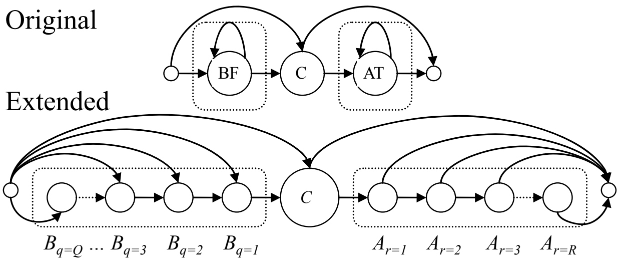


Fig. 5. Original and Extended Metamodels

The extended version of the Metamodels was taken for this work as it was reported to perform better than the original version [2] for the modelling of insertion patterns. Hence, the difference to the original Metamodels is the extension of the insertion states to model the insertion-context associated to each phoneme: the states B_q model the q-th *insertion-before* a phoneme, and A_r the r-th *insertion-after* a phoneme. These $q = 1 \ldots Q$ and $r = 1 \ldots R$ indexes identify the contexts of such insertions, where Q and R represent the length of the contexts. More details about this technique can be found in [2].

4 Optimization Method: Genetic Algorithm

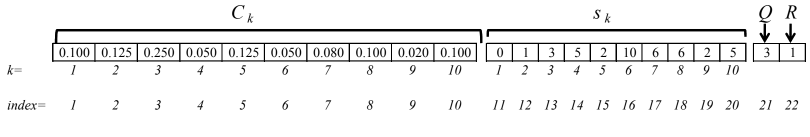
A Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution and generates useful solutions to optimization problems [6]. In a GA, the solutions for an optimization problem receive a penalization score based on their quality or "fitness", which determines their opportunities for reproduction. It is expected that parent solutions of very good quality will produce offsprings (by means of reproduction operators such as crossover or mutation) with similar or better characteristics, improving their fitness after some generations. For our problem, we need to find the following:

- a vector of values C_k for the weights of the K phoneme confusion-matrices used to dynamically obtain a representative matrix for error modelling;
- the grammar scale factor s more suitable to obtain representative phoneme confusion-matrices;
- the length of the insertion-contexts Q and R for the extended Metamodels.

The coding scheme and GA implementation are discussed in the following sections.

4.1 Chromosome Representation and Fitness Evaluation

The chromosome of the GA is presented in Figure 6. The first 10 elements (genes) consist of the weights of C_k, where each one is estimated initially as $C_k = rand(1, 10)/10$ (hence, K=10 phoneme confusion-matrices are considered). Note that these values are not integers, and that the initial set of values for C_k may not sum to 1.0 because each one is estimated individually. Thus, these values must be normalized after the last value is estimated.



	0.100	0.125	0.250	0.050	0.125	0.050	0.080	0.100	0.020	0.100	0	1	3	5	2	10	6	6	2	5	3	1
k=	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*		
index=	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*

Fig. 6. Chromosome representation of the parameters for the Metamodels and the estimation of the dynamic confusion-matrix

From the 10 phoneme confusion-matrices to be used, the first 9 are estimated from the ASR system's phoneme sequences obtained with a phoneme-based (PB) language model. The last one is estimated from the transcription of the word output of the same ASR system, which is obtained with a word-based (WB) language model. Thus, if the following 10 genes correspond to the grammar scale factors $s_k \in$(0,20) (which have integer values) for the estimation of each k-th confusion matrix, the first 9 are for the PB language model, and the last one for the WB language model. Finally, in the last two genes (21, 22) are stored the lengths of the insertion-contexts Q and R ($\in$ (0,10)).

The initial population for the GA consists of 10 individuals, where the first element is the extended metamodel built with the single confusion-matrix approach (hence, genes 1 - 10, 11-20, and 21-22 remain constant with their original values), and the remaining individuals are randomly generated within the range of values specified above.

The fitness value was measured as the Word Recognition Accuracy (%WAcc) obtained by the built Metamodel with the single or dynamic confusion-matrix approach on a "control" speech set (see Section 5).

4.2 Operators: Selection and Reproduction

- **Selection**: How to choose the eligible parents for reproduction was based on the Roulette Wheel and was implemented as follows:
  1. For each of the 10 best individuals in the population, compute its fitness value.
  2. Compute the selection probability for each x_i individual as: $p_i = \frac{f_i}{\sum_{k=1}^{H} f_k}$, where H is the size of the population (sub-set of 10 individuals), and f_i the fitness value of the individual x_i.
  3. Compute the accumulated probability q_i for each individual as: $q_i = \sum_{j=1}^{i} p_j$.
  4. Generate a uniform random number $r \in \{0, 1\}$.
  5. If $r < q_i$, then select the first individual (x_1), otherwise, select x_i such that $q_{i-1} < r \leq q_i$.
  6. Repeat Steps 4 and 5 H times (until all H individuals are selected).
- **Crossover**: Uniform crossover was used for reproduction of parents chosen by the Roulette Wheel method. A template vector of dimension 1×22 was used for this, where each of its elements received a random binary value (0, 1). Offspring 1 is produced by copying the corresponding genes from Parent 1 where the template vector has a value of 0, and copying the genes from Parent 2 where the template vector has a value of 1. Offspring 2 is obtained by doing the inverse procedure. 10 offsprings are obtained by crossover from the 10 individuals in the initial population. This increased the size of the population to 20.
- **Mutation**: The mutation scheme consisted in randomly changing all chromosome values in the best 5 individuals. In this way, a population of 25 individuals is obtained for the GA.
- **Stop Condition**: The algorithm was performed for 20 iterations as it was observed that convergence was stable by that point.

5 Experiments on Dysarthric Speech

5.1 Speech Data and Baseline Recogniser

For the experiments with dysarthric speech, the Nemours database [1] was used. This database consists of a collection of 814 short sentences spoken by 11 american-english speakers (74 sentences per speaker) with varying degrees of dysarthria resulting from either Cerebral Palsy or head trauma (data from only 10 speakers was used as some data is missing for one speaker). The sentences are nonsense phrases that have a simple syntax of the form "the X is Y the Z", where X and Z are monosyllabic nouns (74 in total) and Y is a bisyllabic verb (37 in total) in present participle form (for instance, the phrases "The shin is going the who", "The inn is heaping the shin", etc.).

In addition, data from a reference speaker with normal speech is included. This speaker utters each one of the 740 sentences spoken by the dysarthric speakers. Hence, a baseline speech recogniser was built with this speaker's data. This task was accomplished with the HTK Toolkit [20]. In general, 39 monophone acoustic HMMs were constructed with a standard three state left-right topology with eight mixture components per state. The front-end used 12 MFCCs plus energy, delta and acceleration coefficients. A frame period of 10 msec with a Hamming window of 25 msec and 26 filter-bank channels were used.

To adapt the baseline system to each dysarthric speaker, a common practice when using commercial ASR systems for people with speech disorders, the Maximum Likelihood Linear Regression (MLLR) technique was used [20]. The language models for the baseline, the GA, and the Metamodels, consisted of WB and PB bigrams estimated from all the 740 sentences in the database.

From each dysarthric speaker, 18 randomly selected sentences were used for speaker adaptation and phoneme confusion-matrix estimation. Then, a different set of 18 sentences was selected for fitness evaluation of the GA (control). Finally, the resulting Metamodels of the GA optimization were tested with the remaining 38 sentences.

5.2 Convergence of the GA

Figure 7 shows the mean graph of fitness convergence across 20 executions of the GA and all dysarthric speakers. The Initial Metamodels are those built with the single confusion-matrix approach, the Baseline HMM is the performance of the adapted ASR system, and the Dyn-GA Metamodels are those built with the dynamic confusion-matrix approach optimized with the GA. As it is observed, the Initial Metamodels perform better than the baseline, something that is consistent with the results obtained in [3] and [12]. The Dyn-GA Metamodels increase their performance as the GA iterates, achieving a stable convergence after 10 iterations. It's important to mention that all of the Dyn-GA Metamodels obtained on each of the 20 executions of the GA achieved the same results on the test set. Hence, the results presented in Table 1 were obtained with a single execution of the GA.

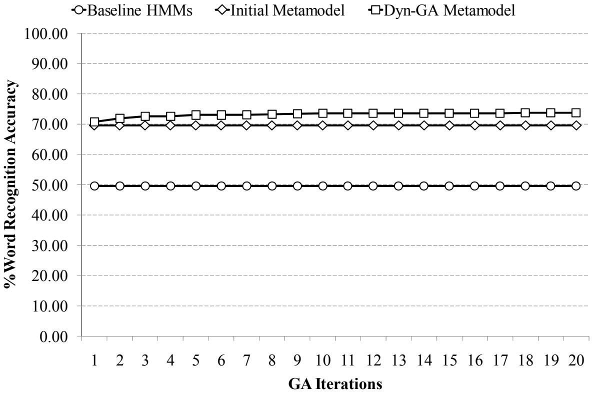


Fig. 7. Convergence of the GA

5.3 Results on Test Data

The mean recognition results across all dysarthric speakers on the test set (380 sentences) are presented in Table 1. A gain of 3.3% (4.5% absolute over the Initial Metamodels performance) is achieved with the Metamodels built with the proposed appoach. This gain was statistically significant as measured by the matched-pairs test described in [5] obtaining a p-value < 0.05.

Table 1. Mean %WAcc across all speakers on the test set

System	%Word Accuracy
Adapted Baseline	54.702
Initial Metamodels	73.406
Dyn-GA Metamodels	76.724

6 Conclusions and Future Work

In this paper we present an approach to obtain phoneme confusion-matrix estimates to improve the performance of Metamodels, a technique for dysarthric ASR. The approach consists of the dynamic estimation of confusion-matrices across different language model restrictions. A single confusion matrix then is estimated from the weighted confusion matrices previously obtained. The weights and other parameters of the Metamodels were optimized by means of a Genetic Algorithm (GA).

As presented in Section 5, with the proposed approach statistically significant gains were achieved over a standard baseline system and Metamodels when tested with a well known dysarthric speech database.

Future work will be aimed at (1) evaluating the performance of the approach with larger language models and other speech databases (TED, WSJ); (2) improving the GA with other operators for reproduction and selection; (3) to apply the GA on other HMM parameters as the number of states and the observation probabilities associated to each HMM state; and (4) to explore the use of the dynamic phoneme confusion-matrices for speaker identification purposes.

References

1. Bunnel, H.T., Polikoff, J.B., Menéndez-Pidal, X., Peters, S.M., Leonzio, J.E.: The Nemours Database of Dysarthric Speech. In: Proc. International Conference on Spoken Language Processing (1996)
2. Caballero, S.O.: Structure Optimization of Metamodels to Improve Speech Recognition Accuracy. In: Proc. of the International Conference on Electronics Communications and Computers (CONIELECOMP 2011), pp. 125–130 (2011)
3. Caballero, S.O., Cox, S.J.: Modelling Errors in Automatic Speech Recognition for Dysarthric Speakers. EURASIP J. Adv. Signal Processing, 1–14 (2009)
4. Caballero, S.O., Cox, S.J.: On the Estimation and the Use of Confusion-Matrices for Improving ASR Accuracy. In: Proc. of Interspeech 2009 (2009)
5. Gillick, L., Cox, S.J.: Some statistical issues in the comparison of speech recognition algorithms. In: Proc. IEEE Conf. on Acoustics, Speech and Signal Processing, pp. 532–535 (1989)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co. (1989)
7. Green, P., Carmichael, J., Hatzis, A., Enderby, P., Hawley, M.S., Parker, M.: Automatic speech recognition with sparse training data for dysarthric speakers. In: Proc. European Conf. on Speech Communication Technology, pp. 1189–1192 (2003)
8. Hamidi, F., Baljko, M., Livingston, N., Spalteholz, L.: CanSpeak: A Customizable Speech Interface for People with Dysarthric Speech. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2010, Part 1. LNCS, vol. 6179, pp. 605–612. Springer, Heidelberg (2010)
9. Hasegawa-Johnson, M., Gunderson, J., Perlman, A., Huang, T.: HMM-based and SVM-based recognition of the speech of talkers with spastic dysarthria. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (2006)
10. Kent, R.D., Wang, Y.T., Duffy, J.R., Thomas, J.E.: Dysarthria associated with traumatic brain injury: speaking rate and emphatic stress. Journal of Communication Disorders 38, 231–260 (2005)
11. Levit, M., Alshawi, H., Gorin, A., Nöth, E.: Context-Sensitive Evaluation and Correction of Phone Recognition Output. In: Proc. European Conference on Speech Communication and Technology, ISCA (2003)
12. Matsumasa, H., Takiguchi, T., Ariki, Y., Li, I.-C., Nakabayash, T.: Integration of Metamodel and Acoustic Model for Dysarthric Speech Recognition. Journal of Multimedia - JMM 4(4), 254–261 (2009)
13. Niu, X., Santen, J.P.: A formant-trajectory model and its usage in comparing coarticulatory effects in dysarthric and normal speech. In: Proc. Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA), pp. 233–236 (2003)
14. Polur, P.D., Miller, G.E.: Effect of high frequency spectral components in computer recognition of dysarthric speech based on a Mel-cepstral stochastic model. Journal of Rehabilitation Research and Development 42(3), 363–372 (2005)
15. Raghavendra, P., Rosengren, E., Hunnicutt, S.: An investigation of different degrees of dysarthric speech as input to speaker adaptive and speaker dependent recognition systems. Aug. & Alt. Communication 17, 265–275 (2001)

16. Rosen, K., Yampolsky, S.: Automatic speech recognition and a review of its functioning with dysarthric speech. Aug. & Alt. Communication 16, 48–60 (2000)
17. Seong, W.K., Park, J.H., Kim, H.K.: Dysarthric Speech Recognition Error Correction Using Weighted Finite State Transducers Based on Context–Dependent Pronunciation Variation. In: Miesenberger, K., Karshmer, A., Penaz, P., Zagler, W. (eds.) ICCHP 2012, Part II. LNCS, vol. 7383, pp. 475–482. Springer, Heidelberg (2012)
18. Strik, H., Sanders, E., Ruiter, M., Beijer, L.: Automatic recognition of dutch dysarthric speech: a pilot study. In: ICSLP, pp. 661–664 (2002)
19. Wu, C.-H., Su, H.-Y., Shen, H.-P.: Articulation-Disordered Speech Recognition Using Speaker-Adaptive Acoustic Models and Personalized Articulation Patterns. ACM Transactions on Asian Language Information Processing (TALIP) 10(2) (2011)
20. Young, S., Woodland, P.: The HTK Book (for HTK Version 3.4). Cambridge University Engineering Department (2006)

Modelling, Aggregation and Simulation of a Dynamic Biological System through Fuzzy Cognitive Maps

Gonzalo Nápoles, Isel Grau, Maikel León, and Ricardo Grau

Centro de Estudios de Informática
Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba
{gnapoles,igrau,mle,rgrau}@uclv.edu.cu

Abstract. The complex dynamics of Human Immunodeficiency Virus leads to serious problems on predicting the drug resistance. Several machine learning techniques have been proposed for modelling this classification problem, but most of them are difficult to aggregate and interpret. In fact, in last years the protein modelling of this virus has become, from diverse points of view, an open problem for researchers. This paper presents a modelling of the protease protein as a dynamic system through Fuzzy Cognitive Maps, using the amino acids contact energies for the sequence description. In addition, a learning scheme based on swarm intelligence called PSO-RSVN is used to estimate the causal weight matrix that characterizes these structures. Finally, an aggregation procedure with previously adjusted maps is applied for obtaining a prototype map, in order to discover knowledge in the causal influences, and simulate the system behaviour when a single (or multiple) mutation takes place.

Keywords: Human Immunodeficiency Virus, Fuzzy Cognitive Maps, Learning Scheme, Aggregation Procedure, Knowledge Discovery, Simulation.

1 Introduction

The Human Immunodeficiency Virus (HIV) is a complex and dynamical *Lentivirus* plaguing humanity harshly causing millions of deaths yearly. Although antiviral drugs are not able to eradicate the HIV, they are designed to inhibit the function of three essential proteins in the virus replication process: protease, reverse transcriptase and integrase. For example, protease inhibitors avoid the maturation of released viral particles by cleaving precursor proteins. However, due to a high mutation rate, this virus is capable to develop resistance to existing drugs, eventually causing the treatment failure. Thus, modelling of resistance mechanism requires the study of viral genome for designing more effective therapies using existing drugs [1].

Resistance testing can be performed either by measuring viral activity in the presence and absence of a drug (phenotypic resistance testing), or by sequencing the viral genes coding for the drug targets (genotypic resistance testing). Genotypic assays are much faster and cheaper, but sequence data provide only indirect evidence of resistance [2], giving a limited knowledge of the HIV behaviour. Therefore, in order to discover more relevant and consistent information, other biological, mathematical, or computational approaches are required.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 188–199, 2013.

Several machine learning techniques have been proposed to predict phenotypic resistance using genotypic information including Neural Networks [3], Recurrent Neural Networks [4], Support Vector Machine regression [5] and Decision Tree classification [6]. These models have achieved good performance in terms of prediction accuracy, but only a few can be interpreted or aggregated for explaining the effect of mutational patterns on the resistance phenotype, that is, the relations among the amino acids and its influence on the resistance.

In this work, the Fuzzy Cognitive Map (FCM) theory is used for modelling the protease protein as a dynamic system. These neural structures are useful to create models that emulate the behaviour of complex and dynamical processes using fuzzy causal relations. Furthermore, a learning scheme based on swarm intelligence is described for adjusting the parameters that characterize the modelled system. Lastly, several previously adjusted FCMs are combined through an aggregation procedure for obtaining a prototype map. This map allows us to: (i) determine patterns in the causal influences of each sequence position on resistance, and (ii) simulate more efficiently the effect of a mutation on the whole system, for five inhibitor drugs.

The rest of the paper is organized as follows: in next Section 2 the theoretical background of FCMs is described, and also the proposal for the modelling problem. In Section 3 we introduce a learning scheme to estimate the FCMs weight matrixes or causal relation values. Section 4 give the experimental setting, provides details and comments about the knowledge discovery process, and also introduces a scheme for simulating the effect of a mutation in the modelled system. Finally, conclusions and further research aspects are discussed in Section 5.

2 Fuzzy Cognitive Maps

Fuzzy Cognitive Maps are an effective Soft Computing technique for modelling and simulation of dynamic systems. They were introduced by Kosko [7], who enhanced cognitive maps with fuzzy reasoning. FCMs combine some aspects of Fuzzy Logic and Neutral Networks in their representation schemes: the heuristic and common sense rules of Fuzzy Logic with the learning heuristics of the Neural Networks. These structures are represented as directed graphs with feedback, consisting of nodes and weighted arcs (see Figure 1). Nodes of the graph stand for the concepts that are used to describe the behaviour of the system, and they are connected by weighted and signed arcs representing the causal relationships that exist among the concepts.

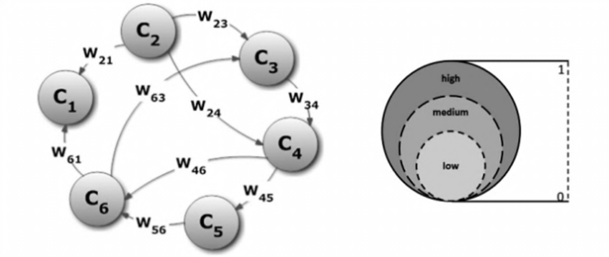


Fig. 1. Simple Fuzzy Cognitive Map. Concept activation level.

The weights of the arcs between concept C_i and concept C_j could be positive which means that an augment in the value of concept C_i leads to the increase of the C_j value, whereas a decrease in the value of concept C_i leads to a reduction of the C_j value; or there is negative causality which means that an increase in the value of concept C_i leads to the decrease of the C_j value and vice versa [8].

The FCMs inference process involves a mathematical model. It consists of a state vector A_{1xn} which includes the values of the n concepts and a weight matrix W_{nxn} which gathers the weights w_{ij} of the interconnections among the n concepts. The value of each concept is influenced by the values of the connected concepts with the appropriate weights and by its previous value. Then, the activation value for each concept is iteratively calculated by the following rule (1):

$$A_i^{(t+1)} = f\left(\sum_{j=1}^{n} w_{ij} A_j^{(t)}\right), i \neq j\,. \tag{1}$$

where t indexes the current time, A_i is the activation level of concept C_i, A_j is the activation level of concept C_j and w_{ij} is the weight of the interconnection between C_i and C_j, whereas *f* is a threshold or normalization function. The new vector shows the effect of the change in a concept value on the whole map.

In order to build an FCM, the knowledge and experience of one expert on the systems operation must be used. The expert determines that concepts that best illustrate the system; where a concept can be a feature of the system, a state or a system variable; identifying which factors are central for the modelling of the system and representing a concept for each one. Moreover, the expert has observed which elements influence other elements, and for corresponding concepts the experts could determine the negative or positive effect of one concept on the others.

It is possible to have better results in the drawing of the FCM, if more than one expert (or knowledge source) is used. This is one advantage over other approaches like Bayesian Networks or Petri Nets. For example, Bayesian Networks is a powerful tool for graphically representing the relationships among a set of variables and for dealing with uncertainties in expert systems [9]; however is not evident for a non-expert in this field how to construct a Bayesian Network, and even more difficult how to combine different Bayesian Networks that describe the same system.

Normally, the aggregation process for FCMs averages multiple maps and their corresponding causal descriptions as well as much their dynamics, producing a new augmented FCM. Therefore, the user (or an automated aggregation procedure) can combine multiple weighted FCMs into a single averaged FCM by the simple artifice of adding their scaled and augmented adjacency edge matrix. In fact, the strong law of large numbers ensures that the sample averaged of even quantized or rounded-off FCM will converge with probability one to the underlying but often unknown FCM that generates these matrix realization [10].

Supported by these ideas an important conclusion came out: the FCMs are a powerful methodology that can be used for modelling statics as well as dynamics systems, avoiding many of the knowledge extraction problems which are usually presented in other systems. Moreover, the FCMs aggregation capabilities provide a useful tool for discover knowledge from multiple knowledge sources.

2.1 Modelling Protease Protein through Fuzzy Cognitive Maps

The main proposition in this work is to model the protease protein as a dynamic system using FCMs theory. This automatic modelling allow us to predict the drug resistance from a given protease mutation sequence and to interpret the causal relations among the amino acids and its influence on the resistance.

Protease sequence is defined by 99 amino acids where each one is characterized by the contact energy; the contact energy [11] of an amino acid is a numerical descriptor that corresponds to the tridimensional structure of proteins and it is calculated statistically from a large number of diverse sequences. So, for modelling this, each sequence position (amino acid) is taken as a map concept, and another extra concept for the resistance is also defined. In addition, we establish causal relationships among all amino acids, and between each one of them and the resistance concept. This proposal is supported by the fact that exist relations among not necessarily adjacent positions of the sequence; where a change in the contact energy of a specific amino acid (to be considered as a mutation) could be relevant in the drug resistance.

To predict the resistance concept (phenotype) from predictor concepts (genotype) means to solve, for each drug, the related sequence classification problem; where the activation value for each predictor is taken as its normalized contact energy (in the range [0,1]). Following this logic, the resistance concept should be considered as a binary concept (1-resistent, 0-susceptible) defined by a drug-specified cut-off. The causal relationships among concepts are defined by a causal weight matrix (W_{nxn+1}). This augmented matrix defines, for each amino acid AA_i, the interaction with the others (AA_j), and additionally characterizes the causal interaction between each descriptor AA_i and the resistance concept (denoted as R). In this model we assume that self-connections are not allowed in the graph, which means that i≠j. Figure 2 shows the general scheme for the modelling process through FCM theory.

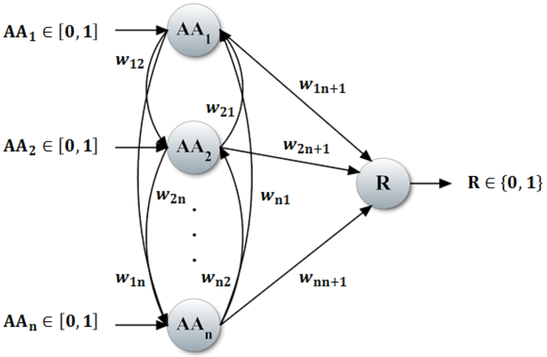


Fig. 2. General scheme for modelling protease through FCM theory

Although modelling a FCM using automatic approaches is relevant from diverse point of views, could be complex to find the parameters that defines the system (e.g. the causal weight matrix), since the knowledge from experts is not enough. Moreover, is often difficult to handle knowledge from different sources of information. For this reason, in the next Section 3 a bioinspired learning method for estimating these parameters using historical data is presented.

3 Weight Matrix Estimation Using Particle Swarm Optimization

Problems associated with automatic modelling of a FCM encourage researchers to work on automated or semi-automated computational methods for learning FCM structures using historical data. Semi-automated methods still require a relatively limited human intervention, whereas fully automated approaches are able to compute a FCM model solely based on historical data [12]. In this Section is proposed an automated scheme to estimate the causal weight matrix using a swarm intelligence based meta-heuristic, particularly Particle Swarm Optimization (PSO).

The PSO algorithm is a stochastic technique that simulates the social behaviour observed in groups or swarms of biological individuals [13]. This meta-heuristic is based on the principle that intelligence does not lie in individuals but in the collective, allowing for the solution of complex optimization problems from a distributed point of view, without centralized control in a specific individual. Each organism (particle) adjusts its position by using a combination of an attraction to the best solution that they individually have found, and an attraction to the best solutions that any particle has found [14], imitating those who have a better performance.

The particles are interpreted as possible solutions for the optimization problem and are represented as points in n-dimensional search space. In the case of standard PSO, each particle (X_i) has its own velocity (V_i) bounded by a maximum value (V_{max}), a memory of the best position it has obtained (P_i) and knowledge of the best solution found in its neighbourhood (G). In the search process the particles adjust their positions according to the following equations (2) and (3):

$$V_i^{(k+1)} = \chi\left(V_i^{(k)} + c_1 r_1 \left(P_i - X_i^{(k)}\right) + c_2 r_2 \left(G - X_i^{(k)}\right)\right) . \tag{2}$$

$$X_i^{(k+1)} = X_i^{(k)} + V_i^{(k+1)} . \tag{3}$$

where k indexes the current generation, c_1 and c_2 are positive constants, r_1 and r_2 are random numbers with uniform distribution on the interval $[0, 1]$, whereas χ represents the constriction coefficient introduced by Clerc et al. [15] and it can be expressed in terms of c_1 and c_2 as shown in (4):

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \text{ and } \varphi = c_1 + c_2 , \varphi > 4 . \tag{4}$$

In this model, the swarm is often attracted by sub-optimal solutions when solving complex multimodal problems, causing premature convergence of the algorithm and swarm stagnation [16]. Once particles have converged prematurely, they continue converging to within extremely close proximity of one another so that the global best and all personal bests are within one minuscule region of the search space, limiting the algorithm exploration. For this reason, we use in this paper a variant of constricted

PSO called PSO-RSVN [17] that uses random samples in variable neighbourhoods for dispersing the particle swarm whenever a premature convergence state is detected, offering an escaping alternative from local optima.

The causal weight matrix estimation could be modelled as an optimization problem where each agent of the swarm represents a possible weight matrix; in this design the particles will encoded as vectors; following equation (5) shows this idea:

$$\vec{\mathrm{x}} = (\mathrm{w}_{11}, \dots, \mathrm{w}_{1\mathrm{n}}, \mathrm{w}_{1\mathrm{n}+1}, \mathrm{w}_{21}, \dots, \mathrm{w}_{2\mathrm{n}}, \mathrm{w}_{2\mathrm{n}+1} \dots, \mathrm{w}_{\mathrm{n}1}, \dots \mathrm{w}_{\mathrm{nn}}, \mathrm{w}_{\mathrm{nn}+1})\,. \tag{5}$$

As mentioned, PSO-RSVN can be used to learn FCM structures based on historical data, consisting of a sequence of state vectors that leads to a desired fixed-point attractor state. In the optimization process the search space is explored, trying to find the best configuration that guarantees a convergence or expected results [18]. This scheme improves the quality of resulting FCM models by minimizing the following objective or heuristic function (6) characterized as:

$$f(\vec{\mathrm{x}}, \Phi) = \sum_{\mathrm{i}=1}^{|\Phi|} |I(\vec{\mathrm{x}}, \Phi_{\mathrm{i}}) - R(\Phi_{\mathrm{i}})|\,. \tag{6}$$

where Φ represents the set of protease mutations or train-cases, whereas *I* is a binary function that calculates the map inference for the i-th mutation using the $\vec{\mathrm{x}}$ as weight matrix. Moreover, other binary function *R* is used for compute the drug resistance (or objective attribute) for the current train-case Φ_{i}. So, this objective function (6) gets an optimal evaluation when the FCM inference process and the resistance criteria stored in the knowledge base are identical, for all mutations.

4 Results and Discussion

In this section we evaluate the proposed model using historical data of five protease inhibitors[1]: Atazanavir (ATV), Indinavir (IDV), Lopinavir (LPV), Nelfinavir (NFV) and Saquinavir (SQV). As mentioned, a stored case represents a protease mutation having 99 attributes or descriptors and a binary class for resistance.

Due to that not all sequence positions which characterize the protein are significant for predicting drug resistance (i.e. not all the amino acids are equally important), in this paper we use a feature selection based on positions that are previously associated with resistance, from both experimental and numerical perspective [20].

Then for each drug we get 100 maps, where the causal weight matrix is estimated using the learning scheme described in the Section 3. Table 1 shows the average prediction accuracy obtained by several machine learning techniques using WEKA toolkit [21]: a Decision Tree (J48), a Bayesian Network (BN) with K2 as learning algorithm, a Support Vector Machine (SVM) with linear kernel function and a

[1] The five knowledge bases were extracted from Stanford HIV Drug Resistance Database [19].

Multilayer Perceptron (MLP). Also a Recurrent Neural Network (RNN) model from [22] was considered. FCM accuracy values were achieved using as parameter settings of the learning scheme: 100 particles, 500 generations, five variable neighbourhoods, and a tolerance (threshold) for the maximum radius of the swarm α = 1.0E-5.

Table 1. Comparison of the performance of FCM against other machine learning techniques in terms of classification accuracy (the best performing algorithm is emphasized in boldface).

Drug	**J48**	**BN**	**SVM**	**MLP**	**RNN**	**FCM**
ATV	0.653	0.678	0.752	0.760	0.744	**0.837**
IDV	0.899	0.894	0.897	0.881	0.875	**0.914**
LPV	0.891	0.897	0.874	0.897	0.897	**0.904**
NFV	0.901	0.859	0.881	0.881	**0.916**	0.859
SQV	**0.928**	0.905	0.905	0.905	0.915	0.848

Although the average prediction accuracies are promising, this work is more focused on discover knowledge in the causal influences of each sequence position on the resistance to five protease inhibitors. Nevertheless, we first need to justify if the further study is reasonable. For this reason, the accuracy is used as a quality measure of the knowledge expressed in the causal relationships.

4.1 Knowledge Discovery and Interpretation

Usually the knowledge obtained from the parameters adjustment process described in Section 3 represents the criterion of a single expert. However, from the FCM theory is well-known that the greater number of FCM to combine, the stronger the knowledge represented in the resultant map. In this paper we use the FCM aggregation process as an effective knowledge discover tool, since this procedure is able to naturally formalize a consistent knowledge from multiple information sources.

Then, for finding patterns in the causal relations between each amino acid and the resistance concept (called direct relations) a prototype map is obtained. First, an aggregation procedure over obtained adjusted maps takes place. This procedure only involves the causal weight matrixes, since the maps that describe the behaviour of protease protein for a specific drug, always have a fixed number of amino acids. Thus, the combined map is characterized by the same concepts that define all individual maps, and by the aggregation of all causal relationship. To aggregate all causal weight matrixes into a single matrix (denoted as Ψ), several operators might be applied. In this paper we use the average operator for obtaining the aggregated weight matrix, using the knowledge of the adjusted weight matrixes, as shown in (7):

$$\Psi_{ij} = \left(w_{ij}^{1} + w_{ij}^{2} + \cdots + w_{ij}^{n}\right)/n\,. \tag{7}$$

After the aggregation process, we take only a portion of the prototype map that corresponds with the knowledge of the direct relations. This portion is related to the causal patterns that explain the effect of a punctual mutation on the resistance. Tables 2, 3, 4, 5 and 6 show the discovered patterns in each prototype map; where a positive value ($\Psi_{ij} > 0$) means that exist a positive causality, a negative value ($\Psi_{ij} < 0$) means that there is a negative causality and a neutral value ($\Psi_{ij} \approx 0$) means that there is no significant causality between the amino acid and the resistance.

Table 2. Amino acid causal influences over resistance for ATV

AA$_{10}$	**AA$_{20}$**	**AA$_{24}$**	**AA$_{32}$**	**AA$_{33}$**	**AA$_{36}$**	**AA$_{46}$**	**AA$_{47}$**	**AA$_{48}$**	**AA$_{50}$**
-0.161	**0.128**	**-0.123**	**0.044**	**-0.097**	**-0.006**	**-0.053**	**-0.059**	**0.324**	**0.542**
AA$_{53}$	**AA$_{54}$**	**AA$_{63}$**	**AA$_{71}$**	**AA$_{73}$**	**AA$_{82}$**	**AA$_{84}$**	**AA$_{88}$**	**AA$_{90}$**	**AA$_{93}$**
-0.008	**0.123**	**-0.178**	**0.264**	**0.106**	**-0.117**	**-0.233**	**0.104**	**-0.448**	**0.231**

Table 3. Amino acid causal influences over resistance for LPV

AA$_{10}$	**AA$_{20}$**	**AA$_{24}$**	**AA$_{32}$**	**AA$_{33}$**	**AA$_{36}$**	**AA$_{46}$**	**AA$_{47}$**	**AA$_{48}$**	**AA$_{50}$**
-0.283	**0.050**	**0.046**	**0.101**	**0.027**	**0.184**	**0.187**	**0.085**	**0.006**	**0.109**
AA$_{53}$	**AA$_{54}$**	**AA$_{63}$**	**AA$_{71}$**	**AA$_{73}$**	**AA$_{82}$**	**AA$_{84}$**	**AA$_{88}$**	**AA$_{90}$**	**AA$_{93}$**
-0.009	**-0.118**	**0.276**	**0.208**	**0.011**	**-0.047**	**-0.024**	**0.096**	**-0.011**	**0.116**

Table 4. Amino acid causal influences over resistance for IDV

AA$_{10}$	**AA$_{20}$**	**AA$_{24}$**	**AA$_{32}$**	**AA$_{33}$**	**AA$_{36}$**	**AA$_{46}$**	**AA$_{47}$**	**AA$_{48}$**	**AA$_{50}$**
-0.644	**0.335**	**0.088**	**0.312**	**0.038**	**-0.290**	**0.338**	**0.344**	**0.444**	**0.292**
AA$_{53}$	**AA$_{54}$**	**AA$_{63}$**	**AA$_{71}$**	**AA$_{73}$**	**AA$_{82}$**	**AA$_{84}$**	**AA$_{88}$**	**AA$_{90}$**	**AA$_{93}$**
0.231	**-0.534**	**-0.111**	**0.348**	**0.190**	**-0.456**	**-0.388**	**0.097**	**-0.218**	**0.170**

Table 5. Amino acid causal influences over resistance for NFV

AA$_{10}$	**AA$_{20}$**	**AA$_{30}$**	**AA$_{36}$**	**AA$_{46}$**	**AA$_{48}$**	**AA$_{50}$**	**AA$_{53}$**	**AA$_{54}$**
-0.456	**0.366**	**0.366**	**0.120**	**0.284**	**0.216**	**0.106**	**0.147**	**-0.309**
AA$_{63}$	**AA$_{71}$**	**AA$_{73}$**	**AA$_{77}$**	**AA$_{82}$**	**AA$_{84}$**	**AA$_{88}$**	**AA$_{90}$**	**AA$_{93}$**
-0.095	**0.150**	**0.133**	**0.313**	**-0.186**	**-0.065**	**0.012**	**-0.366**	**0.007**

Table 6. Amino acid causal influences over resistance for SQV

AA_{10}	AA_{20}	AA_{36}	AA_{46}	AA_{48}	AA_{50}	AA_{53}	AA_{54}
0.474	**-0.414**	**-0.074**	**-0.167**	**-0.411**	**-0.311**	**-0.571**	**0.218**
AA_{63}	**AA_{71}**	**AA_{73}**	**AA_{82}**	**AA_{84}**	**AA_{90}**	**AA_{93}**	
0.105	**-0.141**	**-0.114**	**-0.100**	**0.384**	**0.605**	**0.004**	

Analysing the causal patterns for SQV described in Table 6, some conclusions came out. (i) For the direct relations from AA_{10}, AA_{54}, AA_{63}, AA_{84} and AA_{90} a positive causality exists; which means that when a mutation in one of these amino acids takes place, the resistance value is proportionally affected. More specifically, if one of these amino acids mutes to another one with higher (lower) contact energy then the resistance value proportionally increases (decreases) with a factor Ψ_{ij}. (ii) For the direct relations from AA_{20}, AA_{46}, AA_{48}, AA_{50}, AA_{53}, AA_{71}, AA_{73} and AA_{82} a negative causality exists; which means that when a mutation in one of these amino acids occurs, the resistance value is reverse proportionally affected; if one of these amino acids mutes to another one with higher (lower) contact energy, then the resistance value proportionally decreases (increases), with a factor Ψ_{ij}. (iii) For the direct relations from AA_{84} and AA_{90} there are no significant causal influences on the resistance concept. However, a mutation on these amino acids can be significant for the whole map, since the amino acids interactions should be considered.

Likewise can be drawn conclusions about causal patterns in the direct relations of the other four modelled drugs. These mutational patterns characterize the effect of a single mutation on the resistance value, allowing to the drugs designer a more competent comprehension of this complex biological system.

For example, from the study of the tridimensional structure of proteins is known that all enzymes act on molecules of a substance called *substrate*. This interaction occurs in a specific portion of the enzyme: the *active centre*; where the tridimensional structure matches with a specific substrate. Then the knowledge from discovered patterns could be useful to reconstruct the chemical reactions that occur in the active centre of the protein between the enzyme and its substrate, which is relevant information for creating inhibitor drugs for specific enzymes. More specifically, the characterization of this active centre could help on designing protease inhibitors that have more compatibility with the active centre than the natural substrates, since when the drug is docked to the active site it avoid the enzymatic reaction. This inhibition mechanism can lead to produce a progeny of immature virions that normally are unable to infect the immunologic cells.

4.2 Scheme for Simulating Mutations through FCMs

As mentioned, the knowledge discovery process discussed in previous Section 4.1 is capable to determine important patterns in the dynamic of the HIV, more specifically in the behaviour of the protease protein. This knowledge characterizes the interaction of each amino acid on resistance concept, but the intrinsic relations among all others amino acids are omitted. Nevertheless, these patterns may be difficult to handle when multiple mutations take place. For example: (a) How is affected the drug resistance when simultaneous variations on the amino acids contact energy take place? (b) How is affected the resistance concept when two strong mutations occur and their causal weight are equally strong but opposite? (c) How is affected the resistance concept by concepts that have been stimulated by others mutations?

Typically, the FCMs involve in their representation schemes the information to simulate the effects of a single (or multiple) mutation for the whole system. Thus, the drug expert could predict the resistance for a specific drug, whenever mutations take place in the protein. From the FCMs point of view, once the map is trained with historical data associated to the resistance to a specific drug and a prototype map is obtained, the behaviour of the system could be observed by simply varying the concept value of the corresponding contact energy, which induces the mutation over the wild sequence. Afterwards the inference mechanism is triggered and the resistance concept is examined: if the resistance value is greater that a known threshold, then the mutation will be resistant to a specific drug. Following this logic, drug designers could elaborate information of the protease nature, taking into account the mutational patterns and their effects on the global system.

On other hand, the simulation scheme for the mutations could also be used to find other patterns associated with the dynamic of the HIV. For example, is well-known that the virus normally mutes the sequence positions that are not relevant for the structure and function of the enzyme, that is, the positions which are not related with the active centre or the catalytic process. So, based on the proposal it is possible to formulate a combinatorial problem that allows us to generate high-resistant (or high-susceptible) mutations. These artificial sequences are an important knowledge source for characterizing each sequence position, from the resistance point of view.

In fact, the real success of inhibitors of HIV protease is its revolutionary effect on the current drug design. Particularly, the HIV protease is currently the model for the design of structure-based drug, including recent attempts to design vaccines against this virus. However, other drugs for inhibiting the functions of the other proteins (e.g. integrase or reverse transcriptase) are also required. For this reason, future work will be focused on extending the knowledge discovery process for other proteins.

5 Conclusions

In this paper was proposed a novel FCMs based scheme for modelling the dynamic of the protease protein; the behaviour of these complex systems is very difficult to be described by a precise mathematical model, due to its high mutability. Although several machine learning techniques have been proposed for predicting the drug resistance, most of them are difficult to interpret. However, through FCMs theory it is easier and more practical to represent it in a graphical way, showing the causal relationships between all the protein positions and the resistance.

We showed the benefits of the application of a learning method inspired in the PSO meta-heuristic to fine-tune the causal relationships of a modelled FCM. This supervised weight adaptation methodology adjusts the maps parameters based only on historical data, without human intervention. Experimental results showed that the FCMs based model and the learning method proposed get promising classification accuracies against other approaches reported in the literature.

The aggregation process of several FCMs into a single prototype map was used for knowledge discovering. This procedure is useful to find patterns in the causal influences of each sequence position on resistance to five inhibitor drugs, and reveal that frequently when a mutation in one amino acid takes place, the resistance value is proportionally affected. Moreover, a scheme for simulating the protein behaviour was also proposed, complementing the knowledge discovery process. This knowledge could help drug designers to have more comprehension of this complex biological system. So, this proposal is a first approximation to modelling and studying the dynamic of the HIV proteins on the drug resistance problems, through FCMs. As future work, the knowledge discovery process will be extended and deeper assessed, also other descriptors must be used in order to characterize the amino acids.

Acknowledgments. We would like to thank Prof. Dr. Yovani Marrero-Ponce from Unit of Computer-Aided Molecular Biosilico Discovery and Bioinformatics Research, UCLV, for fruitful discussions and critical reading of the manuscript.

References

1. Prosperi, M., Ulivi, G.: Evolutionary Fuzzy Modelling for Drug Resistant HIV-1 Treatment Optimization. SCI, vol. 82, pp. 251–287. Springer, Heidelberg (2008)
2. Beerenwinkel, N., et al.: Computational methods for the design of effective therapies against drug resistant HIV strains. Bioinformatics 21, 3943–3950 (2005)
3. Drăghici, S., Potter, R.: Predicting HIV drug resistance with neural networks. Bioinformatics 19, 98–107 (2003)
4. Bonet, I., García, M.M., Saeys, Y., Van de Peer, Y., Grau, R.: Predicting Human Immunodeficiency Virus (HIV) Drug Resistance Using Recurrent Neural Networks. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4527, pp. 234–243. Springer, Heidelberg (2007)
5. Beerenwinkel, N., et al.: Geno2pheno: interpreting genotypic HIV drug resistance tests. IEEE Intelligence System 16, 35–41 (2001)
6. Beerenwinkel, N., et al.: Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype. Proceedings of the National Academy of USA 99, 8271–8276 (2002)
7. Kosko, B.: Fuzzy Cognitive Maps. International Journal of Man-Machine Studies 24, 65–75 (1986)
8. Kosko, B.: Neural Networks and Fuzzy systems, a dynamic system approach to machine intelligence. Prentice-Hall, Englewood Cliffs (1992)
9. Castillo, E.: Expert Systems and Probabilistic Network Models. Springer (2003)

10. León, M., Nápoles, G., Rodriguez, C., García, M.M., Bello, R., Vanhoof, K.: A Fuzzy Cognitive Maps Modeling, Learning and Simulation Framework for Studying Complex System. In: Ferrández, J.M., Álvarez Sánchez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2011, Part II. LNCS, vol. 6687, pp. 243–256. Springer, Heidelberg (2011)
11. Miyazawa, S., Jernigan, R.L.: Contacts energies Self-Consistent Estimation of Inter-Residue Protein Contact Energies Based on an Equilibrium Mixture Approximation of Residues. PROTEINS: Structure, Function, and Genetics 34, 49–68 (1999)
12. McMichael, J.M., et al.: Optimizing Fuzzy Cognitive Maps with a Genetic Algorithm. In: AIAA 1st Intelligent Systems Technical Conference, Chicago, Illinois (2004)
13. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, Australia, pp. 1942–1948 (1995)
14. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), Honolulu, HI, pp. 120–127 (2007)
15. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)
16. Kennedy, J., Russell, C.E.: Swarm Intelligence. Morgan Kaufmann Publishers (2001)
17. Nápoles, G., Grau, I., Bello, R.: Particle Swarm Optimization with Random Sampling in Variable Neighbourhoods for Solving Global Minimization Problems. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Engelbrecht, A.P., Groß, R., Stützle, T. (eds.) ANTS 2012. LNCS, vol. 7461, pp. 352–353. Springer, Heidelberg (2012)
18. León, M., Nápoles, G., García, M.M., Bello, R., Vanhoof, K.: Two Steps Individuals Travel Behavior Modeling through Fuzzy Cognitive Maps Pre-definition and Learning. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part II. LNCS, vol. 7095, pp. 82–94. Springer, Heidelberg (2011)
19. Stanford HIV Drug Resistance Database, http://hivdb.stanford.edu
20. Woods, M., Carpenter, G.A.: Neural Network and Bioinformatic Methods for Predicting HIV-1 Protease Inhibitor Resistance. CAS/CNS Technical Report 2007-004 (2007)
21. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann Publishers (2011)
22. Bonet, I.: Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial. PhD Thesis (2008)

Artificial Neural Network for Optimization of a Synthesis Process of γ-Bi_2MoO_6 Using Surface Response Methodology

Guillermo González-Campos[1,*], Edith Luévano-Hipólito[2],
Luis Martin Torres-Treviño[2], and Azael Martinez-De La Cruz[1,2]

[1] Facultad de Ingeniería Mecánica y Eléctrica, UANL
San Nicolás de los Garza, Nuevo León, México
guillermogc53@gmail.com
[2] Centro de Innovación, Investigación y Desarrollo en Ingeniería y Tecnología,
UANL Apodaca, Nuevo León, México
{edithlue,luis.torres.ciidit,azael70}@gmail.com

Abstract. In this work an artificial neural network was utilized in order to optimize the synthesis process of γ-Bi_2MoO_6 oxide by co-precipitation assisted with ultrasonic radiation. This oxide is recognized as an efficient photocatalyst for degradation of organic pollutants in aqueous media. For the synthesis of γ-Bi_2MoO_6 three variables were considered, the exposure time to ultrasonic radiation, calcination time and temperature. The efficiency of photocatalysts synthesized was evaluated in the photodegradation of rhodamine B (rhB) under sun-like irradiation. A set of experimental data were introduced into the neural network, a multilayer type perceptron with a back-propagation learning rule was used to simulate the results by modifying one of the three input variables and observing the efficiency of photocatalysts using besides a response surface methodology.

Keywords: Artificial neural network, heterogeneous photocatalysis, γ-Bi_2MoO_6, response surface.

1 Introduction

The use of an artificial neural network to predict results from different kind of processes is a technique that is being utilized more and more, the reason is because it has been positioned as good technique with a very good precision in the results gained after a good training process and also with a good quantity of experimental data. The wide application that it has on several sciences, makes that the neural networks are being used and cited in several works recently.

1.1 Heterogeneous Photocatalysis

During the last decade the heterogeneous photocatalysis has been positioned as an emergent technology to solve environmental problems [1, 2]. Some of the most

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 200–210, 2013.

remarkable advantages of the heterogeneous photocatalysis is that it is an environment friendly technology due to the use of inert and innocuous photocatalysts (typically ceramic oxides), and the gather of the free solar energy to activate the photocatalytic process. In the same way, the heterogeneous photocatalysis is able to abate the concentration of pollutants until concentration levels where other technologies fail. For these reasons the use of heterogeneous photocatalysis has been applied to the purification of wastewaters and indoor air. The high photocatalytic activity under UV radiation, low cost, and stability to corrosion processes of the anatase polymorph of TiO_2 has positioned this material as the photocatalyst par excellence for commercial applications. However, its relative wide energy band gap (Eg) of 3.2 eV limits further applications of the material in the visible-light region ($\lambda > 390$ nm). In the search of semiconductor materials with photocatalytic activity under visible-light irradiation, important efforts have been carried out in the last decade. For example, the TiO_2 anatase polymorph has been doped with nitrogen in order to increase its absorption in the visible range [3]. In another approach, several authors have proposed alternative oxides than traditional TiO_2 with high photocatalytic activity under visible-light irradiation such as; $In_{1-x}Ni_xTaO_4$, $CaIn_2O_4$, $InVO_4$, $BiVO_4$ y γ-Bi_2MoO_6 [4-9].

Bismuth molybdate has an Aurivillius type structure that has atoms of molybdenum in octahedral positions which is interesting from the viewpoint of photocatalysis. The efficiency of photocatalytic process depends on several parameters such as crystallinity, surface morphology, surface area and heat treatment. These parameters can be modified depending on the experimental conditions of synthesis to obtain the semiconductor oxide. In this paper the organic dye rhodamine B was selected as a contaminant to study the efficiency of γ-Bi_2MoO_6 as a photocatalyst. The parameter used to measure the efficiency of photocatalytic process was the half-life time of the organic dye, which consists of the time that the concentration of the contaminant is reduced to half.

1.2 Artificial Neural Network

Artificial Neural Network (ANN) is a technique inspired by biological neuron processing. It is has a wide application on several sciences for time series forecasting, pattern recognition and process control. Training of the neural networks is sensitive to the number of neurons in the hidden layer. A better performance of the neural network in fitting data can be reached when is involved a high number of neurons. However too many neurons in the hidden layer may result in the over fitting. Finding the optimum number of neurons in the hidden layer is a crucial job during the network design [10].

The neural network has some desirable characteristics like robustness and precision in the approximation. Linear regression is ideal to explain the relationship between variables (neural networks are black boxes); nevertheless, the precision of the regression prediction is lower than neural networks in some cases [11]. Currently the work of Ghanbary Fatemeh et al [12] has used an ANN for optimization of the synthesis of compound TiO_2, another efficient photocatalyst. Recently Filofteia-Laura

Toma et al [13] have reported the use of neural network to measure the efficiency of TiO_2 powder catalysts in relation to the degradation of pollutants in the air such as NO_x which confirms that ANN is a tool that predicts the elimination of these contaminants. In different works, several authors reported the use of ANN to study the process of water purification by homogeneous and heterogeneous catalysis [14], showing an effective and simple approach that describes the behavior of this type of decontamination processes. More recently A. R. Soleymani, et al. (2011) [15], reported the use of an ANN with back-propagation learning algorithm using an activation function of type "tansig" and "pureline" in the middle layer and output, respectively, obtaining efficient results to reproduce experimental data and predict the behavior of dye removal processes by photocatalysis.

1.3 Multi-perceptron Neural Network with Back Propagation Learning Rule

A multilayer perceptron with a back-propagation learning rule was used. This structure consists of a set of layers composed of single neurons. Every neuron executes a simple set of operations to emulate the functions of natural neurons in a simplistic way. The integration of the inputs signal captured by dendrites is made by the soma of the neuron. The artificial neuron consists of a summation of the inputs signal represented by elements of a vector. The weight that regulates every input signal emulates the function of the synapses in natural neurons. When the integrated signal is superior of a threshold level, an output excitation signal is generated and this is dispersed through other neurons by the axon. Otherwise the output signal is inhibited. This operation is represented by a non linear activation function (per example a sigmoidal one). The weights represent the synapse and determine the performance of the neural network. The learning rule is used to update these weights. Several learning rules can be considered; however, the back propagation learning rule with momentum is commonly used [16, 17]. The following equations illustrate the implementation of the back propagation learning rule considering only a multilayer perceptron with *I* inputs, *M* neurons in the inner layer and one output Y_N. The activation function used is a sigmoid one, shown in equation 1.

$$f_a(x) = 1/(1 + e^{-x}) \ . \tag{1}$$

Two operations are defined, the activation and the training of the neural network. The objective is the adjustment of the weights of every layer (matrices *WM* and *WO*) to minimize the error between the desired response (represented in vector Y_D) and the response of the neural neuron (Y_N) considering the same input vector (*X*). The first step is the activation of the inner neurons that it is made using equation 2 and 3.

$$N_M(m) = \textstyle\sum_{i=1}^{I} W_M(m,i)X(i) \, . \tag{2}$$

$$R_M(m) = \ f_a\big(N_M(m)\big) \, . \tag{3}$$

N_M is total of neurons of the inner layer, where *m* is equal to 1:*M*, and *M* is the number of neurons in the inner layer. *I* is equal to 1:*I*, where *I* is the number of inputs of the neural network. There are only one output Y_N so the activation of a single neuron is required (equation 4 and 5).

$$N_O = \sum_{m=1}^{M} W_o(m) R_M(m). \tag{4}$$

$$Y_N = f_a(N_O). \tag{5}$$

R_M is a vector that represents the output of every neuron in the inner layer. The training of the neural network is made by the adjustment of the inner (*WM*) and output weights (*WO*). First, the error between the desired output Y_D and the output of the neural network Y_N is calculated (equation 6). The adjustment of the weight is made using equation 7. The calculation of the momentum requires the values of the weights before the adjustment so this calculation is made using equation 8.

$$\delta_O = \left(Y_{D_p} - Y_{N_p}\right) f'_a(N_O). \tag{6}$$

$$W_O(m) \leftarrow W_O(m) + \eta\delta_O N_M(m) + \alpha W_{OA}(m). \tag{7}$$

$$W_{OA}(m) = \eta\delta_O N_M m. \tag{8}$$

The internal weights require an error signal calculated using equation 9. This calculation requires a derivative of the activation function (equation 10). The inner weights are actualized using equation 11 and the weights before the adjustment is required using equation 12.

$$\delta_m(m) = f'_a\big(Nm(m)\big)\delta_O W_O(m). \tag{9}$$

$$f'_a(x) = f_a(x)\big(1 - f_a(x)\big). \tag{10}$$

$$W_M(m,i) \leftarrow W_M(m,i) + \eta\delta_m X(i) + \alpha W_{MA}(m,i). \tag{11}$$

$$W_{MA}(m) = \eta\delta_m(m) X(i). \tag{12}$$

Assuming that a pair of patterns (X_p, Y_{Np}) is considered. Y_{Np} is the output of the neural network generated by the input pattern X_p and the desired output pattern is YD_p. Following equations (6) to (12) per a pair of patterns p, all the weights are adjusted until an error measurement is minimized (equation 13)

$$E_G = (1/2)\sqrt{\left(\sum_{p=1}^{N_P} \left(\mathrm{Y}_{\mathrm{D_p}} - \mathrm{Y}_{\mathrm{N_p}}\right)^2\right)} . \tag{13}$$

2 Experimental Descriptions

2.1 Synthesis of γ-Bi_2MoO_6

The γ-Bi_2MoO_6 material was synthesized by co-precipitation assisted with ultrasonic radiation. For this purpose two aqueous solutions were prepared. In the first one, 9.1508 g of $Bi(NO_3)_3{\cdot}5H_2O$ [Aldrich, 99.99%] were dissolved in 100 mL of diluted HNO_3. The second one was prepared by dissolving 1.7359 g of $(NH_4)_6Mo_7O_{24}{\cdot}4H_2O$

[Aldrich, 99.99%] in 100 mL of distilled water. The bismuth nitrate solution was added dropwise to the molybdate solution with a vigorous stirring. When the mix was reached, the pH of the solution was adjusted at 3 by using a 2M NH_4OH solution. Afterward, the solution was exposed to ultrasonic radiation in a water bath at 60°C (70W, 42 kHz). The resulting yellow suspension was maintained at 100°C to promote a slow evaporation of the solvent. The yellow powder obtained after this thermal treatment was used as precursor of γ-Bi_2MoO_6. A slow thermal treatment of 10°C/min in air at 300, 350, 400, 450 and 500°C was employed to obtain polycrystalline powders of γ-Bi_2MoO_6.

2.2 Experimental Setup

Structural characterization of the γ-Bi_2MoO_6 samples was carried out by X-ray powder diffraction using a Bruker D8 advanced diffractometer with CuKα radiation (λ=1.5418Å) coupled with a Vantec high speed detector and Ni filters. X-ray diffraction data of the samples was collected in the 2θ range of 10-70o with a scan rate of 0.05o 0.05 s−1. The photochemical reactor employed in this work consisted of a borosilicate glass beaker surrounded by a water-jacket to maintain the reaction temperature at 25°C ± 1°C. A Xe lamp of 5,600 K was used as visible-light source. The γ-Bi_2MoO_6 photocatalytic activity was evaluated on the degradation reaction of rhB in water. In a glass beaker, 250 mL of rhB solution (5 mg L-1) containing 250 mg of photocatalyst were put in ultrasonic bath for 5 minutes to eliminate aggregates. In order to be sure that adsorption desorption equilibrium of the dye on the catalyst surface was reached, the solution was kept in the dark for 1 h. After this time, the light source was turned on. During reaction, samples of 6 mL were taken at different time intervals and then separated through double centrifugation (4000 rpm, 20 min). The supernatant solution was decanted and the rhB concentration was determined through its absorption maximum band (554 nm) by using a UV-Vis spectrophotometer (Perkin Elmer Lambda 35).

2.3 Modeling Responses

The process to acquire many experimental data sometimes gets difficult to reach, because the price and time of each experiment is high. The experimental data set consists of 12 experiments in order to synthesize a material with the highest photocatalytic activity (Table1). For this purpose, the materials are tested as photocatalyst in the photocatalytic degradation of rhodamine and the half-life time of the organic dye in solution is measured to determine the activity of the photocatalyst. The input layer has 3 variables; exposition time to ultrasonic radiation, calcination time and calcinations temperature. We experiment with a large range of number of neurons in the hidden layer and the number of neurons that yielded the best results were using 8 neurons, thus the hidden layer has 8 neurons and the output layer is half-life time (Figure 1). We used a constant learning parameter 0.25 and a moment of 0.5, also this parameters were the best configuration that we got, resulting in the small mean square error of 0.004,381 in 50,000 epochs.

Table 1. Data of the experimental design

Sample	Ultrasound exposure (Minutes)	Calcination time (Minutes)	Calcination temperature (°C)	Half-life time (Minutes)
1	60	96	300	40.7
2	60	96	400	76.8
3	60	96	450	49.8
4	60	96	500	85.8
5	120	96	300	58.1
6	120	96	350	38.7
7	120	96	400	75.3
8	120	96	450	63.8
9	120	96	500	52.6
10	180	24	100	104.1
11	180	96	450	101.4
12	180	96	500	193.1

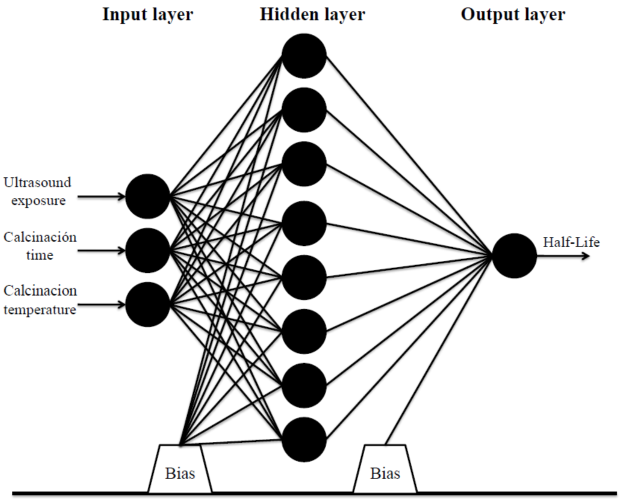


Fig. 1. Architecture of ANN to model the half-life time

The training curve is shown in the Figure 2. The response of the neural network considering training and validation data is shown in Figure 3. The validation data includes the rest of the patterns not used in the training process.

2.4 Test and Analysis

The first part of the tests was simulated half-life times values with the same synthesis parameters obtained in the set of experimental data (Table 2). Then, the data were simulated as constant by adjusting the time of ultrasound exposure for 3 types of different values: 60, 120 and 180 minutes (Tables 3, 4 and 5) to generate the corresponding response surface (Figures 4, 5 and 6) and the results were analyzed.

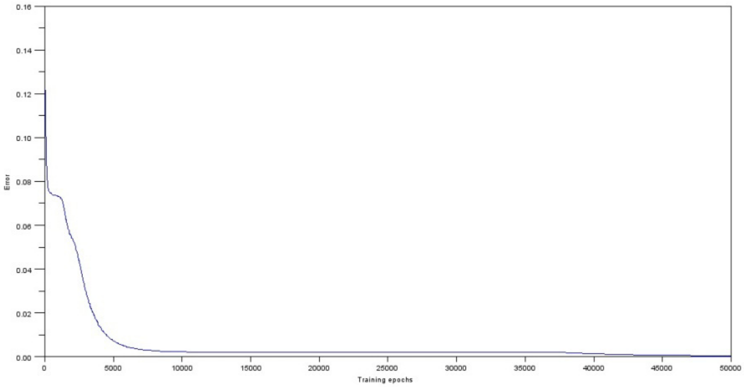


Fig. 2. Training curve (Source: Scilab 5.3.1)

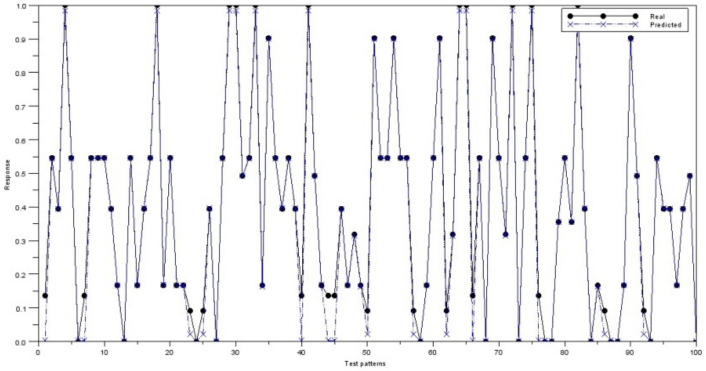


Fig. 3. Response of the neural network versus real data (Source: Scilab 5.3.1)

Table 2. Expected response

Sample	Ultrasound exposure (Minutes)	Calcination time (Minutes)	Calcination temperature (°C)	Expected half-life time (Minutes)	Real half-life time (Minutes)
1	60	96	300	40.9	40.7
2	60	96	400	70.6	76.8
3	60	96	450	79.92	49.8
4	60	96	500	86.9	85.8
5	120	96	300	45.2	58.1
6	120	96	350	52.17	38.7
7	120	96	400	61.1	75.3
8	120	96	450	71.2	63.8
9	120	96	500	86.3	52.6
10	180	24	100	104	104.1
11	180	96	450	102.1	101.4
12	180	96	500	188.4	193.1

Table 3. Expected response with 60 minutes of ultrasound exposure

Ultrasound exposure (Minutes)	Calcination time (Minutes)	Calcination temperature (°C)	Expected half-life time (Minutes)
60	96	300	40.9
60	96	400	70.6
60	96	450	79.92
60	96	500	86.9
60	96	300	40.9
60	96	350	70.6
60	96	400	70.6
60	96	450	79.92
60	96	500	86.9
60	24	100	41.37
60	96	450	79.92

Table 4. Expected response with 120 minutes of ultrasound exposure

Ultrasound exposure (Minutes)	Calcination time (Minutes)	Calcination temperature (°C)	Expected half-life time (Minutes)
120	96	300	45.2
120	96	400	61.1
120	96	450	71.2
120	96	500	86.3
120	96	300	45.2
120	96	350	52.17
120	96	400	61.1
120	96	450	71.2
120	96	500	86.3
120	24	100	40
120	96	450	71.2

Table 5. Expected response with 180 minutes of ultrasound exposure

Ultrasound exposure (Minutes)	Calcination time (Minutes)	Calcination temperature (°C)	Expected half-life time (Minutes)
180	96	300	39.9
180	96	400	50
180	96	450	102.1
180	96	500	188.4
180	96	300	39.9
180	96	350	42.1
180	96	400	50
180	96	450	102.1
180	96	500	188.4
180	24	100	104
180	96	450	102.1

3 Results and Discussion

Response surfaces were generated to study the behavior response of the half-life time, in the x-axis is the calcination temperature, on the vertical axis is the calcination time and the response surface is the half-life time divided into 12 ranges with a resolution of 10 units. The response surface with a constant value of 60 minutes of ultrasound

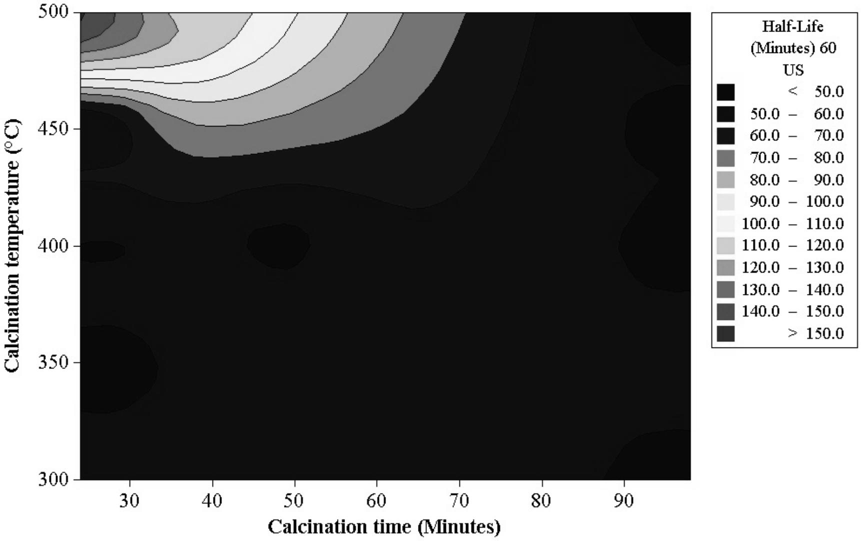


Fig. 4. Response surface with 60 minutes of ultrasonic exposure (Source: Minitab 15)

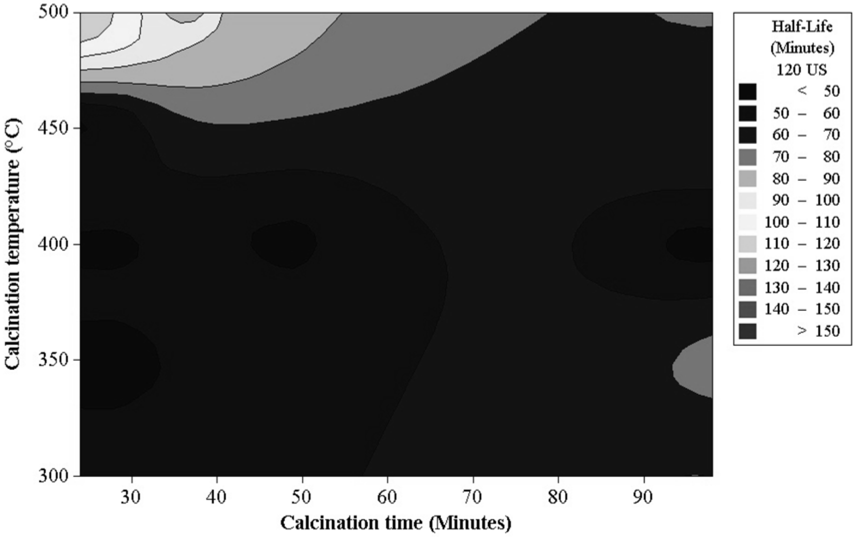


Fig. 5. Response surface with 120 minutes of ultrasonic exposure (Source Minitab 15)

radiation (Fig. 4), there are regions where the half-life time may be lower at higher temperatures and lower calcination time will result in a longer half-life time, which is not the best purpose, but we can see areas where we get the best half-life time.

The response surface of 120 minutes of ultrasound (Fig. 5) shows that the ranges of half-life time are higher than 60 minutes of ultrasound, moreover was identified that increasing the calcination time will gradually increase the half-life time.

And finally in the response surface of 180 minutes of ultrasound (Fig. 6), is completely lost the aim of obtaining a low value of half-life time compared with other response surfaces.

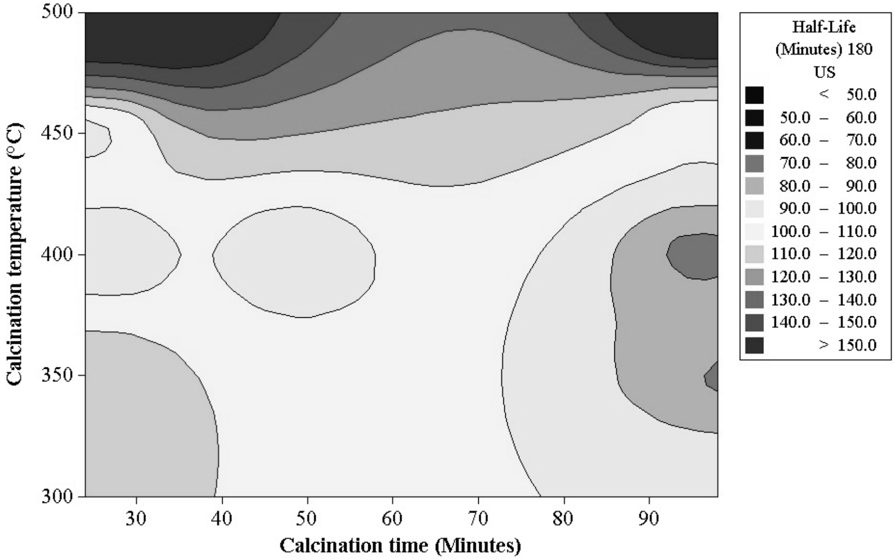


Fig. 6. Response surface with 180 minutes of ultrasonic exposure (Source: Minitab 15)

4 Conclusions

In spite of the few quantity of experimental information that we used to predict the results, we got a very good precision in the data and also in the response surfaces generated from the simulated data. The use of this type of artificial neural network and one or more statistical tools is a good method to optimize the synthesis of photocatalysts, thus resulting in savings of time and materials for implementation. We can also show that an ANN could be used in different areas and it has a wide application on several sciences. In the case to optimize other properties of the γ-Bi_2MoO_6, such as surface area, particle size, the existence of a single phase, crystallinity, so on, we will have to make another model which together will be get better results in the synthesis of nanoparticles of γ-Bi_2MoO_6.

Acknowledgements. We wish to thank to the Universidad Autónoma de Nuevo León (UANL) for its invaluable support through the project PAICYT 2010 and to CONACYT for supports the projects CB-167018 and CIAM-148138.

References

1. Chong, M.N., Jin, B., Chow, C.W.K., Saint, C.: Recent developments in photocatalytic water treatment technology: a review. Water Research 44, 2997 (2010)
2. Laufs, S., Burgeth, G., Duttlinger, W., Kurtenbach, R., Maban, M., Thomas, C., Wiesen, P., Kleffmann, J.: Conversion of nitrogen oxides on commercial photocatalytic dispersion paints. Atmospheric Environment (2010), doi:10.1016/j.atmosenv.2010.03.038
3. Yang, S., Gao, L.: New method to prepare nitrogen-doped titanium dioxide and its photocatalytic activities irradiated by visible light. Journal of the American Ceramic Society 87, 1803–1805 (2004)
4. Jain, R., Mathur, M., Sikarwar, S., Mittal, A.: Removal of the Hazardous Dye Rhodamine B through Photocatalytic and Adsorption Treatments. Journal of Environmental Management 85(4), 956–964 (2007)
5. Zou, Z.G., Ye, J.H., Sayama, K., Arakawa, H.: Direct Splitting of Water under Visible Light Irradiation with an Oxide Semiconductor Photocatalyst. Nature 414, 625–627 (2001)
6. Tang, J.W., Zou, Z.G., Ye, J.H.: Effects of Substituting Sr2+ and Ba2+ for Ca2+ on the Structural Properties and Photocatalytic Behaviors of CaIn2O4. Chemistry of Materials 16(9), 1644–1649 (2004)
7. Zou, Z., Ye, J., Sayama, K., Arakawa, H.: Photocatalytic and Photophysical Properties of a Novel Series of Solid Photocatalysts, Bi2MNbO7 (M=Al3+,Ga3+ and In3+). Chemical Physics Letters 333(1-2), 57–62 (2001)
8. Kudo, A., Omori, K., Kato, H.: A Novel Aqueous Process for Preparation of Crystal Form-Controlled and Highly Crystalline BiVO4 Powder from Layered Vanadates at Room Temperature and Its Photocatalytic and Photophysical Properties. Journal American Chemistry Sociecty 121(49), 11459–11467 (1999)
9. Kato, H., Hori, M., Konta, R., Shimodaira, Y., Kudo, A.: Construction of ZScheme-Type Heterogeneous Photocatalysis Systems for Water Splitting into H2 and O2 under Visible Light Irradiation. Chemistry Letters 33(10), 1348–1349 (2004)
10. Dutta, S., Parsons, S.A., Bhattacharjee, C., Bandhyopadhyay, S., Datta, S.: Development of an artificial neural network model for adsorption and photocatalysis of reactive dye on TiO2 surface. Expert Systems with Applications 37, 8634–8638 (2010)
11. Torres-Treviño, L.M., Reyes-Valdes, F.A., López, V., Praga-Alejo, R.: Multi-objective optimization of a welding process by the estimation of the Pareto optimal set. Expert Systems with Applications 38, 8045–8053 (2011)
12. Ghanbary, F., Modirshahla, N., Khosravi, M., Behnajady, M.A.: Synthesis of TiO2 nanoparticles in different thermal conditions and modeling its photocatalytic activity with artificial neural network. Journal of Environmental Sciences 24(4), 750–756 (2012)
13. Toma, F.L., Guessasma, S., Klein, D., Montavon, G., Bertrand, G., Coddet, C.: Neural computation to predict TiO2 photocatalytic efficiency for nitrogen oxides removal. Journal of Photochemistry and Photobiology A: Chemistry 165, 91–96 (2004)
14. Khataee, A.R., Kasiri, M.B.: Artificial neural networks modeling of contaminated water treatment processes by homogeneous and heterogeneous nanocatalysis. Journal of Molecular Catalysis A: Chemical 331, 86–100 (2010)
15. Soleymani, A.L., Saien, J., Bayat, H.: Artificial neural networks developed for prediction of dye decolorization efficiency with UV/K2S2O8 process. Chemical Engineering Journal 170, 29–35 (2011)
16. Haykin, S.: Neural networks: A comprehensive foundation. Prentice Hall (1999)
17. Freeman, J.A., Skapura, D.M.: Neural networks: Algorithms, applications, and programming techniques. Addison-Wesley (1991)

Recurrent Neural Control of a Continuous Bioprocess Using First and Second Order Learning

Carlos-Román Mariaca-Gaspar, Julio-César Tovar Rodríguez,
and Floriberto Ortiz-Rodríguez

Department of Communication and Electronic Engineering,
Higher School of Mechanic and Electrical Engineering - Zacatenco,
National Polytechnic Institute
Av. Juan de Dios Batiz s/n, Mexico D.F., 07738, Mexico
{cmariacag,jctovar,flortiz}@ipn.com

Abstract. The propose of this paper is to introduce a new Kalman Filter based in a Recurrent Neural Network topology (KFRNN) and a recursive Levenberg-Marquardt (L-M) algorithm. Such algorithm is able to estimate the states and parameters of a highly nonlinear continuous fermentation bioprocess in noisy environment. The control scheme is direct adaptive and also contains feedback and feedforward recurrent neural controllers. The proposed control scheme is applied for real-time identification and control of continuous stirred tank bioreactor model, taken from the literature, where a fast convergence, noise filtering and low mean squared error of reference tracking were achieved.

Keywords: Kalman filter, recurrent neural network, recurrent trainable neural network controller, real-time backpropagation learning, recursive Levenberg-Marquardt learning, real-time direct adaptive neural control, continuous stirred tank reactor bioprocess.

1 Introduction

In a FFNN the signals are transmitted only in one direction, starting from the input layer, subsequently through the hidden layers to the output layer, which requires applying a tap delayed global feedbacks and a tap delayed inputs to achieve a Nonlinear Autoregressive Moving Average (NARMAX) neural dynamic plant model. A new Kalman Filter Recurrent Neural Network (KFRNN) topology and the recursive Backpropagation (BP) type learning algorithm in vector-matrix form was derived and its convergence was studied, [4], [5]. But the recursive BP algorithm, applied for KFRNN learning, is a gradient descent first order learning algorithm which not permits to augment the precision and to accelerate the learning. So, the main contribution in this paper is to use a second order learning algorithm for the KFRNN like the Levenberg-Marquardt (L-M) algorithm [6]-[8], in order to obtain a fast convergence, avoid noise and the effect of unmodeled dynamics as much as possible. The KFRNN with L-M learning will be applied for a Continuous Stirred Tank Reactor (CSTR)

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 211–222, 2013.

model, [9], identification and control. In [10], [11] a comparative study of linear, nonlinear and neural-network-based adaptive controllers for a CSTR is done. The papers proposed to use the neuro-fuzzy and adaptive nonlinear control systems design applying FFNNs (multilayer perceptron and radial basis functions NN). The proposed control gives a good adaptation to the nonlinear plants dynamics, better with respect to the other methods of control. The application of KFRNNs together with the recursive L-M could avoid these problems improving the learning and the precision of the plant states estimation.

2 Topology and Learning of the RTNN

This section is dedicated to the topology, the BP and the L-M algorithms of the Recurrent Trainable Neural Network, [5], [15]-[17] learning. The RTNN topology could be obtained from the KFRNN topology removing the output local and global feedbacks. The RTNN was used as a feedback and feedforward controller in direct adaptive neural control scheme.

2.1 RTNN Topology

The RTNN topology is described by the following vector-matrix equations:

$$X(k+1) = A_1 X(k) + BU(k) \tag{1}$$

$$B = [B_1; B_0];\; U^T = \left[U_1^{\,T}; U_2^{\,T}\right] \tag{2}$$

$$A = \text{block-diag}(A_i),\; |A_i| < 1 \tag{3}$$

$$Z_1(k) = G[X(k)] \tag{4}$$

$$C = [C_1; C_0];\; Z^T = \left[Z_1^{\,T}; Z_2^{\,T}\right] \tag{5}$$

$$V(k) = CZ(k) \tag{6}$$

$$Y(k) = F[V(k)] \tag{7}$$

Where: X, Y, U are vectors of state, output, and augmented input with dimensions N, L, (M+1), respectively, Z is an (L+1) –dimensional input of the feedforward output layer, where Z_1 and U_1 are the (Nx1) output and (Mx1) input of the hidden layer; the constant scalar threshold entries are Z_2 = -1, U_2 = -1, respectively; V is a (Lx1) pre-synaptic activity of the output layer; the super-index T means vector transpose; A is (NxN) block-diagonal weight matrix; B and C are [Nx(M+1)] and [Lx(N+1)]-augmented weight matrices; B_0 and C_0 are (Nx1) and (Lx1) threshold weights of the hidden and output layers; F[·], G[·] are vector-valued tanh(·) or sigmoid(·) –activation functions with corresponding dimensions. Equation (3) represents the local stability condition imposed on all blocks of A. The dimension of the state vector X of the RTNN is chosen using the simple rule of thumb which is: N=L+M.

3 Backpropagation RTNN Learning

Following the same procedure as for the KFRNN, it was possible to derive the following updates for the RTNN weight matrices:

$$E_1(k) = F'[Y(k)]E(k); E(k) = Y_p(k) - Y(k) \tag{8}$$

$$\Delta C(k) = E_1(k) Z^T(k) \tag{9}$$

$$E_3(k) = G'[Z(k)]E_2(k); E_2(k) = C^T E_1(k) \tag{10}$$

$$\Delta B(k) = E_3(k) U^T(k) \tag{11}$$

$$\Delta A(k) = E_3(k) X^T(k) \tag{12}$$

$$\Delta vA(k) = E_3(k) \oplus X(k) \tag{13}$$

Where ΔA, ΔB, ΔC are weight corrections of the learned matrices A, B, and C, respectively; E, E_1, E_2, and E_3 are error vectors; X is a state vector; F'(·) and G'(·) are diagonal Jacobean matrices, whose elements are derivatives of the tanh activation functions . Equation (12) represents the learning of the full feedback weight matrix of the hidden layer. Equation (13) gives the learning solution when this matrix is diagonal vA, which is the present case. The initial values of the weight matrices during the learning are chosen as arbitrary numbers inside a small range. The stability of the RTNN model used as a direct controller is assured by the activation functions [-1, 1] bounds and by the local stability weight bound condition given by (3). The stability of the RTNN movement around the optimal weight point has been proved by one theorem (see the thesis of Mariaca [13] and the paper of Nava, [14]).

4 Stability Proof of the KFRNN BP Learning (Theorem of Stability)

The *Theorem of Stability of the BP RTNN* used as a system controller is proven in the PhD thesis of Mariaca [13], so it shall be given here in brief. Let the RTNN with Jordan Canonical Structure is given by equations (1)-(7). Under the assumption of RTNN identifiability made, the application of the BP learning algorithm for A(·), B(·), C(·), in general matricial form, (8)-(13) without momentum term, and the learning rate η (k) (here it is considered as time-dependent and normalized with respect to the error) are derived using the following Lyapunov function:

$$L(k) = L_1(k) + L_2(k) \tag{14}$$

Where: L_1 (k) and L_2 (k) are given by:

$$L_1(k) = \tfrac{1}{2} e^2(k)$$

$$L_2(k) = tr\left(\tilde{W}_{A(k)} \tilde{W}^T_{A(k)}\right) + tr\left(\tilde{W}_{B_{(k)}} \tilde{W}^T_{B(k)}\right) + tr\left(\tilde{W}_{C(k)} \tilde{W}^T_{C(k)}\right)$$

Where

$$\tilde{W}_{A(k)} = \hat{A}_{(k)} - A^*,\ \ \tilde{W}_{B(k)} = \hat{B}_{(k)} - B^*,\ \ \tilde{W}_{C(k)} = \hat{C}_{(k)} - C^*$$

are vectors of the estimation error and (A^*, B^*, C^*) and $(\hat{A}_{(k)}, \hat{B}_{(k)}, \hat{C}_{(k)})$ denote the ideal neural weight and the estimate of neural weight at the k-th step, respectively, for each case.

Let us define: $\psi_{\max} = \max_k \left\|\psi(k)\right\|$, and $\vartheta_{\max} = \max_k \left\|\vartheta(k)\right\|$. Here: $\psi(k) = \partial o(k) / \partial W(k)$, and $\vartheta(k) = \partial y(k) / \partial u(k)$, where W is a vector composed by all weights of the RTNN, used as a system controller, and $\|\cdot\|$ is an Euclidean norm in $\Re^n$.

Then the identification error is bounded, i.e.:

$$L(k+1) = L_1(k+1) + L_2(k+1) < 0 \tag{15}$$

$$DL(k+1) = L(k+1) - L(k) \tag{16}$$

Where the condition for L_1 (k+1) < 0 is that:

$$0 < \eta_{\max} < \frac{2}{\vartheta^2_{\max} \psi^2_{\max}}$$

and for L_2(k+1) < 0, we have:

$$\Delta L_2(k+1) < -\eta_{\max} \left|e(k+1)\right|^2 + \beta(k+1) \tag{17}$$

Note that $\eta_{\max}$ changes adaptively during the learning process of the network and:

$$\eta_{\max} = \max_{i=1}^{3} \{\eta_i\}$$

Where all: the unmodelled dynamics, the approximation errors and the perturbations, are represented by the β-term, and the complete proof of that theorem is given in [13] and the Lemma of convergence of (17) is given by Nava [14].

5 Recursive Levenberg-Marquardt Algorithm of RTNN Learning

The general recursive L-M algorithm of learning, [4], [6]-[8], [17], is given by the equations (18)-(21), where W is the general weight matrix (A, B, C), under modification; Y is the RTNN output vector which depends of the updated weights and the input; E is an error vector; Yp is the plant output vector which is in fact the target vector. Using the RTNN adjoint block diagram, [4], [5], [15]-[17] we could obtain the values of DY for each updated weight propagating D = I through it.

$$W(k+1)=W(k)+P(k)\nabla Y[W(k)]E[W(k)] \tag{18}$$

$$Y[W(k)]=g[W(k),U(k)] \tag{19}$$

$$E^2[W(k)]=\{Y_p(k)-g[W(k),U(k)]\}^2 \tag{20}$$

$$DY[W(k)]=\frac{\partial}{\partial W}g[W,U(k)]\Big|_{W=W(k)} \tag{21}$$

Applying equation (21) for each element of the weight matrices (A, B, C) to be updated, the corresponding gradient components are obtained as follows:

$$DY[C_{ij}(k)]=D_{1,i}(k)Z_j(k) \tag{22}$$

$$D_{1,i}(k)=F_i'[Y_i(k)] \tag{23}$$

$$DY[A_{ij}(k)]=D_{2,i}(k)X_j(k) \tag{24}$$

$$DY[B_{ij}(k)]=D_{2,i}(k)U_j(k) \tag{25}$$

$$D_{2,i}(k)=G_i'[Z_i(k)]C_iD_{1,i}(k) \tag{26}$$

So the Jacobean matrix could be formed as:

$$DY[W(k)]=[DY(C_{ij}(k)),DY(A_{ij}(k)),DY(B_{ij}(k))]$$

Next the given up topology and learning is applied for CSTR system control.

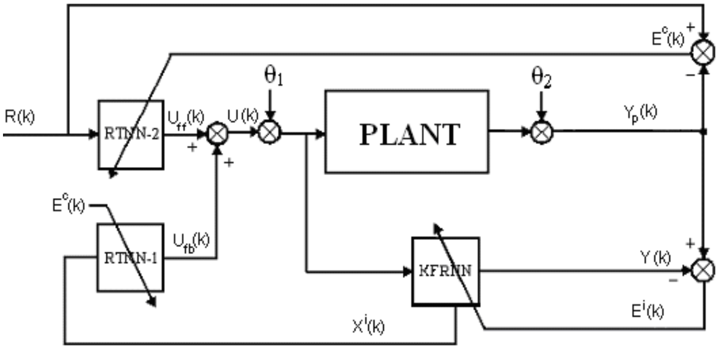


Fig. 1. Block-diagram of the closed-loop neural control

6 Direct Adaptive Neural Control of CSTR System

This part described the direct adaptive CSTR control using KFRNN as plant identifier and RTNN as two plant controllers (feedback and feedforward). The block-diagram of the control system is given on Fig. 1, [5], [15]. Next we described the linearized model of the control system. Let us to write the z-transfer function representations of the plant, the state estimation part of the KFRNN, the feedback / feedforward controllers:

$$W_p(z) = C_p\left(zI - A_p\right)^{-1} B_p \tag{27}$$

$$P_i(z) = \left(zI - A_i\right)^{-1} B_i \tag{28}$$

$$Q_1(z) = C_c\left(zI - A_c\right)^{-1} B_{1_c} \tag{29}$$

$$Q_2(z) = C_c\left(zI - A_c\right)^{-1} B_{2_c} \tag{30}$$

The control systems z-transfer functions (27)-(30) are connected by the following equation, derived from the Fig. 4, and given in z-operational form:

$$Y_p(z) = W_p(z)\left[I + Q_1(z)P_i(z)\right]^{-1} Q_2(z)R(z) + \theta(z) \tag{31}$$

$$\theta(z) = W_p(z)\theta_1(z) + \theta_2(z) \tag{32}$$

Where θ(z) represents a generalized noise term. The RTNN and the KFRNN topologies were controllable and observable, and the BP algorithm of learning was convergent, [5], so the identification and control errors tended to zero:

$$E_i(k) = Y_p(k) - Y(k) \to 0;\ k \to \infty \tag{33}$$

$$E_c(k) = R(k) - Y_p(k) \to 0;\ k \to \infty \tag{34}$$

This means that each transfer functions given by equations (27)-(30) is stable with minimum phase. The closed-loop system is stable and the RTNN-1 feedback controller compensates the plant dynamics. The RTNN-2 feedforward controller dynamics is an inverse dynamics of the closed-loop system one, which assure a precise reference tracking in spite of the presence of process and measurement noises.

7 Description of the CSTR Bioprocess

The CSTR model given in [10], [11], [18] was chosen as a realistic example for application of the KFRNN and the RTNN for solution of system identification and control problems.

The CSTR is described by the following continuous time nonlinear system of ordinary differential equations:

$$\frac{dC_A(t)}{dt} = \frac{Q}{V}\left(C_{Af} - C_A(t)\right) - k_0 C_A(t)\exp\left(-\frac{E}{RT(t)}\right) \tag{35}$$

$$\begin{aligned}\frac{dT(t)}{dt} &= \frac{Q}{V}(T_f - T(t)) + \frac{(-\Delta H)C_A(t)}{\rho C_p}\exp\left(\frac{E}{RT(t)}\right)\\ &+ \frac{\rho_c C_{pc}}{\rho C_p V} Q_c(t)\left[1 - \exp\left(\frac{-hA}{Q_c(t)\rho_c C_{pc}}\right)\right](t_{ef} - T(t))\end{aligned} \tag{36}$$

In this model is enough to know that within the CSTR, two chemicals are mixed and that they react in order or produce a product compound A at a concentration $C_A(t)$, and the temperature of the mixture is T(t). The reaction is exothermic and produces heat which slows down the reaction. By introducing a coolant flow-rate $Q_c(t)$, the temperature can be varied and hence the product concentration can be controlled. Here C_{Af} is the inlet feed concentration; Q is the process flow-rate; T_f and T_{ef} are the inlet feed and coolant temperatures, respectively; all of which are assumed constant at nominal values. Likewise, k_0, E/R, V, ΔH, ρ, C_{pc}, C_p , and ρ_c are thermodynamic and chemical constants related to this particular problem. The quantities Q_{c0}, T_0, and C_{A0}, shown in Table 1, are steady values for a steady operating point in the CSTR. The objective was to control the product compound A by manipulating $Q_c(t)$. The operating values were taken from [7] and [8], where the performance of a NN control system is reported.

Table 1. Parameters and operating conditions of the CSTR

Parameters	Parameters	Parameters
$Q = 100 \quad (L/\min)$	$E/R = 9.95\times10^3 \quad (K)$	$Q_{c0} = 103.41 \quad (L/\min)$
$C_{Af} = 1.0 \quad (mol/L)$	$-\Delta H = 2\times10^5 (cal/mol)$	$hA = 7\times10^5 \quad (cal/\min K)$
$T_f = T_{fC} = 350 \quad (K)$	$\rho\cdot\rho_c = 1000 \quad (g/L)$	$T_0 = 440.2 \quad (K)$
$V = 100 \quad (L)$	$C_pC_{pc} = 1 \quad (cal/gK)$	$k_0 = 7.2\times10^{10} \quad (1/\min)$

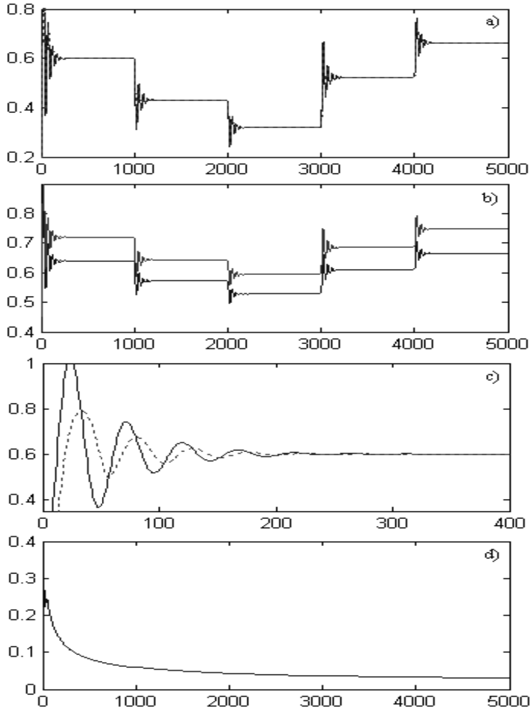


Fig. 2. Graphical results of identification using BP KFRNN learning. a) Comparison of the plant output (continuous line) and KFRNN output (pointed line); b) state variables; c) comparison of the plant output (continuous line) and KFRNN output (pointed line) in the first instants; d) MSE% of identification.

8 Simulation Results

Detailed comparative graphical simulation results of CSTR KFRNN plant identification by means of the BP and the L-M learning are given in Fig.2 and Fig.3. A 10% white noise is added to the plant inputs and outputs and the behavior of the plant identification has been studied accumulating some statistics of the final MSE% (ξ_{av}) for BP and L-M learning, which results are given in Tables 3 and 4 for 20 runs respectively.

The mean average cost for all runs (ε) of KFRNN plant identification and the standard deviation (σ) with respect to the mean value and the deviation (Δ) are presented in Table 2 for BP and L-M algorithm. The computations of these values are done by the following formulas:

$$\varepsilon = \frac{1}{n}\sum_{k=1}^{n} \xi_{av_k}, \sigma = \sqrt{\frac{1}{\mathrm{n}}\sum_{i=1}^{n} \Delta_i^2}\ , \ \Delta = \xi_{av} - \varepsilon \tag{37}$$

The proposed direct adaptive neural control system was applied for the given up CSTR plant. The Fig. 4 compared the reference and the plant output signals applying the L-M algorithm of learning for the KFRNN and both RTNNs. The obtained graphical (Fig. 2 and Fig. 3) and numerical (Table 2) identification results showed that the convergence of the L-M learning algorithm outperformed the BP one. The given control results (Fig. 4) showed a fast reaction and convergence of the L-M learned RNNs.

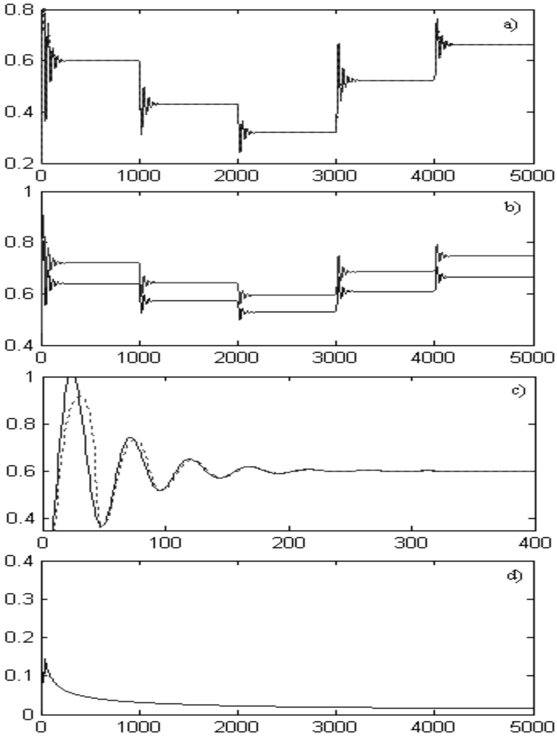


Fig. 3. Graphical results of identification using L-M KFRNN learning. a) Comparison of the plant output (continuous line) and KFRNN output (pointed line); b) state variables; c) comparison of the plant output (continuous line) and KFRNN output (pointed line) in the first instants; d) MSE% of identification.

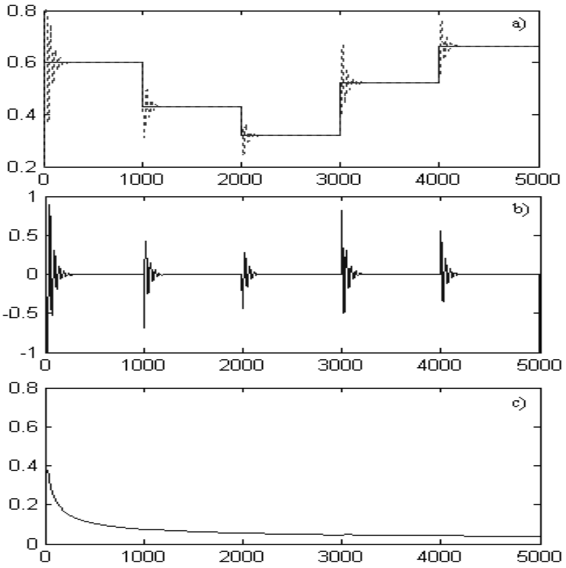


Fig. 4. Detailed graphical simulation results of CSTR plant direct neural control using LM RTNN learning a) comparison between the plant output and the reference signal; b) control signal; c) MSE% of control

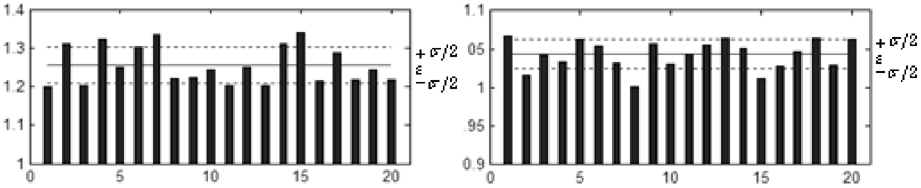


Fig. 5. Comparison between the final MSE% for 20 runs of Identification program: a) using BP algorithm of learning, b) using L-M algorithm of learning

Table 2. Standard deviations and mean average values of identification validation using the BP and L-M algorithms of KF RNN learning

BP algorithm	L-M algorithm
ε = 1.2552	ε = 1.04374
σ = 0.0476	σ = 0.0193

Table 3. MSE% of 20 runs of the identification program using the KFRNN BP algorithm

No	1	2	3	4	5
MSE%	1.2010	1.3104	1.2025	1.3229	1.2494
No	6	7	8	9	10
MSE%	1.3011	1.3356	1.2218	1.2228	1.2439
No	11	12	13	14	15
MSE%	1.2041	1.2501	1.2038	1.3111	1.3399
No	16	17	18	19	20
MSE%	1.2154	1.2872	1.2186	1.2434	1.2189

Table 4. MSE% of 20 runs of the identification program using the KFRNN L-M algorithm

No	1	2	3	4	5
MSE%	1.0665	1.0162	1.0425	1.0340	1.0624
No	6	7	8	9	10
MSE%	1.0533	1.0320	1.0013	1.0575	1.0311
No	11	12	13	14	15
MSE%	1.0431	1.0554	1.0645	1.0517	1.0123
No	16	17	18	19	20
MSE%	1.0284	1.0467	1.0642	1.0287	1.0626

9 Conclusion

The paper proposed a new KFRNN model for systems identification and states estimation of nonlinear plants. The KFRNN is learned by the BP and by the second order recursive learning algorithm of Levenberg-Marquardt. The estimated states of the recurrent neural network model are used for direct adaptive trajectory tracking control systems design. The applicability of the proposed neural control system is confirmed by simulation results with a CSTR plant. The results showed good convergence of both L-M and BP learning algorithms. It was seen that the L-M algorithm of learning is more precise (see Table 2) but more complex than the BP one. The authors thank the support of the Mexican Government (CONACYT, SNI, SIP 20120746, COFAA and PIFI IPN).

References

1. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamic Systems Using Neural Networks. IEEE Trans. Neural Networks 1(1), 4–27 (1990)
2. Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J.: Neural Networks for Control Systems - A Survey. Automatica 28(6), 1083–1112 (1992)
3. Pao, S.A., Phillips, S.M., Sobajic, D.J.: Neural Net Computing and Intelligent Control Systems. International Journal of Control 56, 263–289 (1992)

4. Baruch, I.S., Escalante, S., Mariaca-Gaspar, C.R.: Identification, Filtering and Control of Nonlinear Plants by Recurrent Neural Networks Using First and Second Order Algorithms of Learning. International Journal of Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis, Special Issue on Advances in Neural Networks-Theory and Applications 14 (S!), Part 2, 512–521 (2007) ISSN 1201-3390
5. Baruch, I.S., Mariaca-Gaspar, C.R., Barrera-Cortes, J.: Recurrent Neural Network Identification and Adaptive Neural Control of Hydrocarbon Biodegradation Processes. In: Hu, X., Balasubramaniam, P. (eds.) Recurrent Neural Networks, pp. 61–88. I-Tech/ARS Press, Croatia (2008) ISBN 978-3-902613-28-8
6. Asirvadam, V.S., McLoone, S.F., Irwin, G.W.: Parallel and Separable Recursive Levenberg-Marquardt Training Algorithm. In: Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 129–138 (2002)
7. Ngia, L.S., Sjöberg, J., Viberg, M.: Adaptive Neural Nets Filter Using a Recursive Levenberg-Marquardt Search Direction. IEEE Signals, Systems and Computer 1, 697–701 (1998)
8. Ngia, L.S., Sjöberg, J.: Efficient Training of Neural Nets for Nonlinear Adaptive Filtering Using a Recursive Levenberg Marquardt Algorithm. IEEE Trans. on Signal Processing 48, 1915–1927 (2000)
9. Schmidt, L.D.: The Engineering of Chemical Reactions. Oxford University Press, New York (1998) ISBN 0-19-510588-5
10. Liu, S.-R., Yu, J.-S.: Robust Control Based on Neuro-Fuzzy Systems for a Continuous Stirred Tank Reactor. In: Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, pp. 1483–1488 (2002)
11. Zhang, T., Guay, M.: Adaptive Nonlinear Control of Continuously Stirred Tank Reactor Systems. In: Proceedings of the American Control Conference, Arlington, pp. 1274–1279 (2001)
12. Wan, E., Beaufays, F.: Diagrammatic Method for Deriving and Relating Temporal Neural Networks Algorithms. Neural Computations 8, 182–201 (1996)
13. Mariaca-Gaspar, C.R.: Topologies, Learning and Stability of Hybrid Neural Networks, Applied for Nonlinear Biotechnological Processes, Ph. D. Thesis, Dept. Automatic Control, CINVESTAV-IPN. Mexico (2009)
14. Nava, F.R., Baruch, I.S., Poznyak, A., Nenkova, B.: Stability Proofs of Adavanced Recurrent Neural Networks Topology and Learning, Comptes Rendus. Proceedings of the Bulgarian Academy of Sciences 57(1), 27–32 (2004)
15. Baruch, I.S., Barrera-Cortés, J., Hernández, L.A.: A Fed-Batch Fermentation Process Identification and Direct Adaptive Neural Control with Integral Term. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS (LNAI), vol. 2972, pp. 764–773. Springer, Heidelberg (2004)
16. Baruch, I.S., Georgieva, P., Barrera-Cortes, J., Feyo de Azevedo, S.: Adaptive Recurrent Neural Network Control of Biological Wastewater Treatment. International Journal of Intelligent Systems, Special issue on Soft Computing for Modelling, Simulation and Control of Nonlinear Dynamical Systems 20(2), 173–194 (2005) ISSN 0884-8173
17. Baruch, I.S., Mariaca-Gaspar, C.R.: A Levenberg-Marquardt Learning Applied for Recurrent Neural Identification and Control of a Wastewater Treatment Bioprocess. International Journal of Intelligent Systems 24, 1094–1114 (2009)
18. Lightbody, G., Irwin, G.W.: Nonlinear Control Structures Based on Embedded Neural Systems Models. IEEE Trans. Neural Networks 8, 553–557 (1997)

A Conic Higher Order Neuron Based on Geometric Algebra and Its Implementation

Juan Pablo Serrano Rubio, Arturo Hernández Aguirre,
and Rafael Herrera Guzmán

Center for Research in Mathematics
Computer Science Department
Guanajuato, Guanajuato, Mexico
{jpsr,artha,rherrera}@cimat.mx

Abstract. This paper presents the implementation of a Multilayer Perceptron (MLP) using a new higher order neuron whose decision region is generated by a conic section (circle, ellipse, parabola, hyperbola). We call it the hyper-conic neuron. The conic neuron is defined for the conformal space where it can freely work and take advantage of all the rules of Geometric (Clifford) Algebra. The proposed neuron is a non-linear associator that estimates distances from vectors (points) to decision regions. The computational model of the conic neuron is based on the geometric product (an outer product plus an inner product) of geometric algebra in conformal space. The Particle Swarm Optimization (PSO) algorithm is used to find the values of the weights that properly define some MLP for a given classification problem. The performance is presented with a classical benchmark used in neural computing.

Keywords: Conformal Geometric Algebra, Clifford Algebra, Artificial Neural Networks, Particle Swarm Optimization.

1 Introduction

In recent years several authors have demonstrated that non-linear neurons can generate complex non-linear decision regions with less computational effort than the common sigmoidal neuron. Furthermore, there is a growing interest in sophisticated neural networks that do all computations using the geometric algebra otherwise known as Clifford Algebra. Neural computation with Clifford algebras is based on a generalization of the linear associator (of the MLP) using the geometric product. At the beginning of the 90's the complex valued networks were introduced [1], making possible the extension of the back-propagation algorithm to the complex domain. A few years later the *quaternionic* valued neural network was introduced [2]. The network allowed to interpolate functions of quaternion variables using a smaller number of connections than those needed by the real valued MLP. In the last decade the Hyperbolic Multilayer Perceptron was introduced [3]. The network is based on hyperbolic numbers and 2-D hyperbolic orthogonal transformations that compute the weight propagation function in

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 223–235, 2013.

each neuron. The Clifford algebra is also the basis for other models, such as, Basic Clifford Neuron, Spinor Clifford Neuron, Clifford Multilayer Perceptrons [4], and Geometric Feedforward Neural Networks [5]. Recently, the hyper-sphere neuron was proposed as an extension of the standard neuron used in the MLP (whose activation function is a sigmoid) [6]. For the hyper-sphere neuron the decision surface is a n-dimensional sphere, whereas for the standard neuron the decision surface is a n-dimensional plane. This work presents the hyper-conic neuron (HCN) as a generalization of the hyper-sphere neuron. We also show how to construct a MLP using the HCN and call the new network hyper-conic multilayer perceptron (HC-MLP). The new network combines elements of n-dimensional conic. The training algorithm, based on the Particle Swarm Optimization (PSO), determines the conic function with which every neuron should work as to build the required n-dimensional decision surface. The rest of this paper is organized as follows: In Section 2 we give a brief introduction to geometric algebra, geometric product, and to conformal spaces. Basic PSO algorithm is discussed in Section 3. The proposed HC-MLP and its implementation is presented in Section 4. Results of numerical experiments are presented in Section 5. Comments and conclusions are given in Section 6.

2 Geometric Algebra

Geometric algebra proposes an alternative framework where lines, areas, volumes and hyper-volumes are recognized as geometric objects with magnitude and orientation. The heart of geometric algebra is the geometric product, which is defined as the sum of the inner and outer products. The geometric product of two vectors a,$b \in \Re^n$ is defined by:

$$ab = a \bullet b + a \wedge b. \tag{1}$$

We use the symbol $\bullet$ to denote the inner product or the scalar product. The outer product is the operation denoted by the operator $\wedge$. The properties of the outer product are: antisymmetry, scaling and distributivity. The product $a \wedge b$ is called a bivector. The bivector represents an area between a and b with an orientation. The orientation will be the opposite, if the product is $b \wedge a$, due to the antisymmetry property. Therefore, the outer product is not commutative. The outer product is generalizable in higher dimensions, for example, the outer product $(a \wedge b) \wedge c$ gives a trivector.

Let $\{a_1, ..., a_k\} \subset \Re^n$ be mutually linearly independent vectors, where $k \leq n$. For a vector $b \epsilon \Re^n$,

$$(a_i \wedge a_2 ... \wedge a_k) \wedge b = 0 \tag{2}$$

If and only if b is linearly dependent on $\{a_1, ..., a_k\}$. The outer product of k vectors is called a $k - blade$ and can be written as:

$$A_{<k>} = a_1 \wedge a_2 \wedge ... a_k =: \bigwedge_{i=1}^{k} a_i. \tag{3}$$

The grade $< k >$ of a blade is simply the number of vectors included in the outer product. A basis for the Geometric Algebra Cl_n is given by:

$$1, \{e_i\}, \{e_i \wedge e_j\}, \{e_i \wedge e_j \wedge e_k\}, ..., \{e_1 \wedge e_2 ... \wedge e_n \equiv I\} \tag{4}$$

$\{e_1, ..., e_n\}$ an orthonormal basis of $\Re^n$, such that $e_i \bullet e_j = \delta_{ij}$. I is a hypervolume called pseudoscalar. In the algebra Cl_n any vector can be expressed in terms of its basis. The set of all k-vectors is a vector space, denoted as $\bigwedge^k V^n$, of dimension $\binom{n}{k}$, where $\binom{n}{k} = \frac{n!}{(n-k)!k!}$. The dimension of the Geometric Algebra Cl_n, is $\sum_{k=0}^{n} \binom{n}{k} = 2^n$ is the sum of the subspaces of different grades:

$$Cl_n = \bigwedge^0 V^n \oplus \bigwedge^1 V^n \oplus \bigwedge^2 V^n \oplus ... \oplus \bigwedge^n V^n \tag{5}$$

where $\bigwedge^0 V^n = \Re$, $\bigwedge^1 V^n = V^n$. For example, the Geometric Algebra Cl_3 of a 3-dimensional space has $2^3 = 8$ basis blades.

$$\underbrace{1}_{scalar}, \underbrace{e_1, e_2, e_3}_{vectores}, \underbrace{e_1 \wedge e_2, e_2 \wedge e_3, e_3 \wedge e_1}_{bivectores}, \underbrace{e_1 \wedge e_2 \wedge e_3 \equiv I}_{trivector} \tag{6}$$

A multivector v can be formed as: $v = \alpha_0 + \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 + \alpha_4 e_1 \wedge e_2 + \alpha_5 e_2 \wedge e_3 + \alpha_6 e_3 \wedge e_1 + \alpha_7 e_1 \wedge e_2 \wedge e_3 = \langle v \rangle_0 + \langle v \rangle_1 + \langle v \rangle_2 + \langle v \rangle_3$, where the $\alpha's$ are real numbers and $\langle v \rangle_0 = \alpha_0 \in \bigwedge^0 V^n$, $\langle v \rangle_1 = \alpha_1 e_1 + \alpha_2 e_2 + \alpha_3 e_3 \in \bigwedge^1 V^n$, $\langle v \rangle_2 = \alpha_4 e_1 \wedge e_2 + \alpha_5 e_2 \wedge e_3 + \alpha_6 e_3 \wedge e_1 \in \bigwedge^2 V^n$,$\langle v \rangle_3 = \alpha_7 e_1 \wedge e_2 \wedge e_3 \in \bigwedge^3 V^n$.

2.1 Conformal Space and Conic Section

Points, circles, planes and spheres may be represented by vectors in conformal geometric algebra for a vector space. The inner product of these objects results in a scalar and can be used as a measure for distances. For example, the inner product of two vectors is directly proportional to the square of the Euclidean distance of the Euclidean points $\{p, s\}$ corresponds to the inner product of points in the conformal geometric algebra $\{P, S\}$ multiplied by -2.

$$(s - p)^2 = -2(P \bullet S) \tag{7}$$

The points $\{P, S\}$ can be nulls as consequense of $P \bullet S = 0$. In [7], the authors offer an intuitive way to denote a 2-D conic surface by a 2D-vector $(x, y) \in \Re^2$ transformed to $\Re^6$ as follows:

$$(x, y) \ \in \ \Re^2 \ \mapsto \ \left(\frac{1}{\sqrt{2}}x^2, xy, \frac{1}{\sqrt{2}}y^2, x, y, \frac{1}{\sqrt{2}}\right) \ \in \ \Re^6 \tag{8}$$

The Inner Product Null Space (IPNS) of a vector $A \in \Re^6$ is the set of all those vectors $X \in R^6$ that satisfy $X \bullet A = 0$, when they lie on a conic. On the other hand, given any five points in the plane in a general linear position, there is a unique 2D conic surface passing through them, therefore, it is possible to denote a conic surface by the Outer Product Null Space (OPNS) as: $X_0 \wedge A_{<5>} = X_0 \wedge X_1 \wedge X_2 \wedge X_3 \wedge X_4 \wedge X_5$. In this sense the dual of the result is a vector equivalent to IPNS. The relationship between the IPNS and the OPNS is defined as duality. The dual of a blade $A \in Cl_n$ gives a blade complementing the whole space. The dual of A is written as A^* and is defined as:

$$A^* = A \bullet I^{-1} \tag{9}$$

where I^{-1} is the inverse unit pseudoscalar of Cl_n. The inverse of a blade A is given by:

$$A^{-1} = (-1)^{k(k-1)/2} \frac{A_{<k>}}{\| A_k \|^2} \tag{10}$$

The definition of duality can be used to obtain an orthogonal distance between vectors and blades. Figure 1, shows a blade A which contains two points R and Q with their respective position vectors r and q. Q is the point on the blade nearest to the point P which makes the vector $d = q - p$ perpendicular to A. To get the distance, we use the geometric product $Ad = A \wedge d$ [8]. To isolate the variable d, we multiply both sides of the equation by A^{-1}:

$$A^{-1}Ad = A^{-1}(A \wedge d) \tag{11}$$

$$d = A^{-1}(A \wedge d). \tag{12}$$

Then, we replace $d = p - t - v$ in (12) to get the equation (13):

$$d = A^{-1}(A \wedge (p - r - s)) \tag{13}$$

with the condition $A^{-1}A = 1$. The distance from P to the blade A is $\| d \|$. Equation (13) is used by our proposed hyper-conic neuron for the estimation of the orthogonal distance from any point to its decision surface.

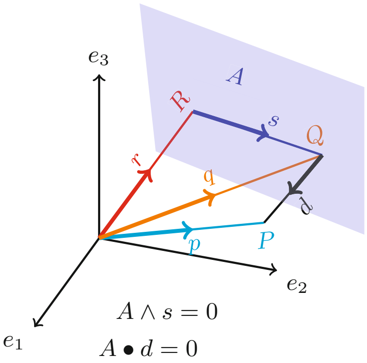


Fig. 1. Distance from a point to a plane

3 Particle Swarm Optimization PSO

A PSO maintains a swarm of particles, where each particle represents a potential solution. Particles are dispersed through a multidimensional search space, where the position of each particle is adjusted according to its own experience and the ones of its neighbors [9]. The social component of the particle velocity update reflects information obtained from all the particles in the swarm. In this sense, the social information of the particle i is the best position found by the swarm, referred as to $\overline{g}_i$. A cognitive component, represents the personal thinking of each particle, referred to the vector $\overline{l}_i$. The velocity $v_{ij}(t+1)$ of particle i in dimension $j = 1, ..., n$ at time step $t+1$, is calculated as follows:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)[l_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[g_{ij}(t) - x_{ij}(t)] \tag{14}$$

where w is the inertia weight, $x_{i,j}(t)$ is the position of particle i in dimension j at time step t, c_1 and c_2 are positive acceleration constants used to scale the contribution of the cognitive and social components respectively, and finally $r_{1j}(t), r_{2j}(t) \sim U(0,1)$ are random values in the range $[0,1]$, sampled from a uniform distribution. These random values introduce a stochastic element to the algorithm. The pseudo-code of the PSO algorithm is illustrated in Figure 2.

Algorithm 1. PSO Algorithm

```
for all particle $i$ do
   //Create and initialize an $n - dimensional$ swarm, $S$
   //position $\overline{x} = x_1, .., x_n$ and velocity $\overline{v} = v_1, .., v_n$.
end for
while stop criteria not met do
   for all particle $i$ do
      if f($S_i.\overline{x}$) < f($S_i.\overline{lb}$) then
         $S_i.\overline{l} = S_i.\overline{x}$ //set personal best $\overline{l} = \{l_1, ..., l_n\}$ as best position found so far
         by the particle.
      end if
      if f($S_i.\overline{x}$) < f($S_i.\overline{gb}$) then
         $S_i.\overline{g} = S_i.\overline{x}$ //set global best $\overline{g} = \{g_1, ..., g_n\}$ as best position found so far by
         the whole swarm.
      end if
   end for
   for all particle $i$ do
      for all dimension $j$ do
         $S_i.v_j(t+1) = w \bullet S_i.v_j(t) + c_1 \bullet r_{1j}(t) \bullet [S_i.l_j(t) - S_i.x_j(t)]$
                     $+ c_2 \bullet r_{2j}(t) \bullet [S_i.g_j(t) - S_i.x_j(t)]$ //update the velocity.
         $S_i.x_j(t+1) = S_i.x_j(t) + S_i.v_j(t+1)$ //update position
      end for
   end for
end while
```

Fig. 2. Particle Swarm Optimization Algorithm

4 Implementation

The standard Perceptron creates decision regions by the composition of hyperplane functions. The proposed hyper-conic neuron, however, creates decision regions by the composition of hyper-conic functions. Figure 3 shows the topology of the HC-MLP. Input signal $\overline{x} = x_1, x_2, ..., x_n$ are propagated through the network from left to right. The symbols $o_1, ...o_k, ..o_l$ denote the outputs of the hidden layer to the output layer. The outputs of the HC-MLP are described by y_1, ..., y_k,...,y_q.

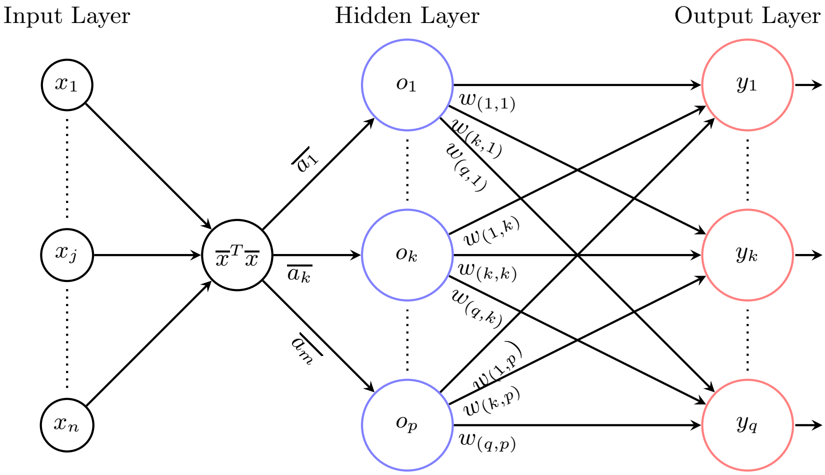


Fig. 3. Hyper-conic Multilayer Perceptron Model

The sigmoid function is used by the proposed model, in the following way:

$$G(x) = \frac{1}{1 + exp(-\lambda \bullet x)}. \tag{15}$$

The output of the neuron k of the hidden layer can be written as a function composition between the non-linear associator and the sigmoid function $o_k = G \circ F(A, \overline{x})$. Inner product null space is implemented by a quadratic equation $\overline{x}^T A \overline{x} = 0$, where matrix A is symmetric of size $(n + 1) \times (n + 1)$.

$$F(A,\overline{x}) = (x_1,\cdots,x_k,\cdots,x_n,1) \bullet \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & a_{1,n+1} \\ a_{1,2} & a_{2,2} & \cdots & a_{2,n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & a_{n,n+1} \\ a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n} & a_{n+1,n+1} \end{pmatrix} \bullet \begin{pmatrix} x_1 \\ \vdots \\ x_k \\ \vdots \\ x_n \\ 1 \end{pmatrix} \tag{16}$$

$$= X \bullet A = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_{(i,j)} a_{(i,j)}, \tag{17}$$

where $X = \overline{x}^T\overline{x}$, the $a_{(i,j)}$ denotes the component of matrix A_k at row i and column j. Therefore, the output of the neuron o_k can be obtained by:

$$o_k = G \circ F(A_k,\overline{x}) \tag{18}$$

$$= \frac{1}{1+exp(-\lambda \bullet F(A,\overline{x}))} \tag{19}$$

$$= \frac{1}{1+exp\left(-\lambda \bullet \left(\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} x_{(i,j)} a_{(i,j)}\right)\right)}. \tag{20}$$

The strategy search of the vector A that best satisfies the above equation for a set of points is usually called an algebraic estimation of a conic. When equation (17) gives zero as a result this indicates that the vector A lies exactly on the conic surface. The aim is to fit a conic which separates the pattern classes by equation (17) which gives a measure distance from the point to the conic surface. The computed distance can be positive or negative; the sign depends on the position of the point with respect to the kind of conic surface.

The output y_k for the neuron k in the output layer is written as a function composition between the linear associator $L(\overline{w}_k,\overline{o},\theta)$ and the sigmoid function:

$$y_k = G \circ L(\overline{w}_k,\overline{o},\theta) \tag{21}$$

$$= \frac{1}{1+exp(-\lambda \bullet L(\overline{w}_k,\overline{o},\theta_k))} \tag{22}$$

$$= \frac{1}{1+exp\left(-\lambda \bullet \left(\sum_{i=1}^{p} w_{(i,k)} o_{(i,k)} - \theta_k\right)\right)} \tag{23}$$

where $\overline{w}_k = (w_1,\ldots,w_k,\ldots,w_\alpha)$ and θ are the weight vector and a bias of the output neuron k in the output layer respectively, $\overline{o} = (o_1,\ldots,o_k,\ldots,o_p)$ is the vector of the outputs of the hidden layer, and p is the number of neurons in the hidden layer.

Similarly, it is possible to implement the outer product null space in each neuron to get a distance between the conic surface and the points. We will use an example to design a 2D-conic surface. The dual will be denoted by $*$. The distance from a point X to the blade A_g can be obtained by the function δ:

$$\delta(X, A_{<g>}) = (X \wedge A_{<g>})^* = (X \wedge A_{<g>})A^{-1}_{<g>}. \tag{24}$$

The last equation can be implemented by the determinant of Eq (27), where the points are mapped from $\Re^2$ to $\Re^6$ by the function:

$$\rho : (x_i, y_i) \ \in \Re^2 \mapsto (x_i^2, x_i \bullet y_i, y_i^2, x_i, y_i, 1). \tag{25}$$

$X_0 \in \Re^6$ is presented as the point $(x_0^2, x_0y_0, y_0^2, x_0, y_0, 1)$, whereas five points $X_1, X_2, \cdots, X_5$ represent the blade $A_{<5>}$.

$$\begin{aligned}
\delta(X_0, A_{<5>})A^{-1}_{<5>} &= (X_0 \wedge X_1 \wedge X_2 \wedge X_3 \wedge X_4 \wedge X_5)^* && (26)\\
&= \det \begin{vmatrix} x_0^2 & x_0y_0 & y_0^2 & x_0 & y_0 & 1 \\ x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & x_3y_3 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & x_4y_4 & y_4^2 & x_4 & y_4 & 1 \\ x_5^2 & x_5y_5 & y_5^2 & x_5 & y_5 & 1 \\ x_6^2 & x_6y_6 & y_6^2 & x_6 & y_6 & 1 \end{vmatrix} && (27)\\
&= x_0^2 a_{1,1} + x_0^2 a_{2,2} + 2x_0y_0a_{1,2} + 2x_0a_{1,3} + 2y_0a_{2,3} + a_{3,3} && (28)\\
&= \overline{x} \bullet \overline{a}. && (29)
\end{aligned}$$

For example, let $\{(5,3), (-5,3), (5,-3), (-5,-3), (4,0)\}$ be a blade $A_{<5>}$, the conic equation can be obtained as:

$$\begin{aligned}
\delta_0((x = x_0, y = y_0), A_{<5>}) &= (X_1 \wedge A_{<5>})A^{-1} && (30)\\
&= \det \begin{vmatrix} x_0^2 & x_0y_0 & y_0^2 & x_0 & y_0 & 1.0 \\ 25.0 & 15.0 & 9.0 & 5.0 & 3.0 & 1.0 \\ 25.0 & \text{-}15.0 & 9.0 & \text{-}5.0 & 3.0 & 1.0 \\ 25.0 & \text{-}15.0 & 9.0 & 5.0 & \text{-}3.0 & 1.0 \\ 25.0 & 15.0 & 9.0 & \text{-}5.0 & \text{-}3.0 & 1.0 \\ 16.0 & 0 & 0 & 4.0 & 0 & 1.0 \end{vmatrix} && (31)\\
&= 32400y^2 - 32400x^2 - 518400 && (32)\\
&= y^2 - x^2 - 16. && (33)
\end{aligned}$$

The function of the equation (33) corresponds to a hyperbolic paraboloid in $\Re^3$, however, when intersects with the Euclidean plane in $\Re^2$ is obtained a hyperbola as illustrated in Figure 4.

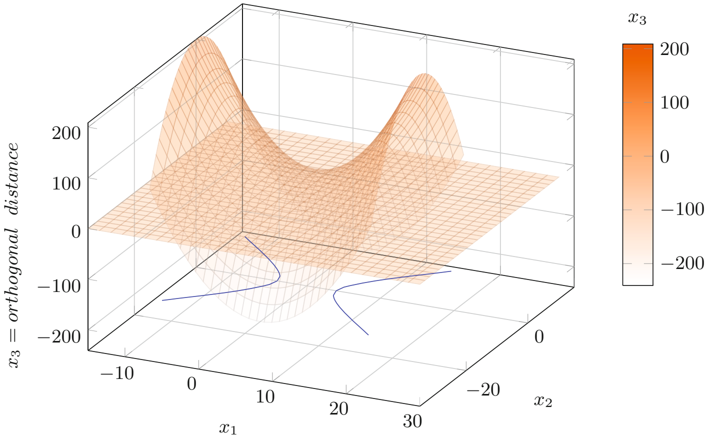


Fig. 4. Intersection between the hyperbolic paraboloid and the Euclidean Plane

Figure 5 shows the contours of the hyperbolic paraboloid which intersects with the Euclidean plane in $\Re^2$. The contour levels of the hyperbolic paraboloid are used as measure distance. The distance is positive when the points (the small circles) are outside the conic, whereas the distance is negative if the points (the small dots) are inside the conic. If the points are on the conic surface, the distance is zero.

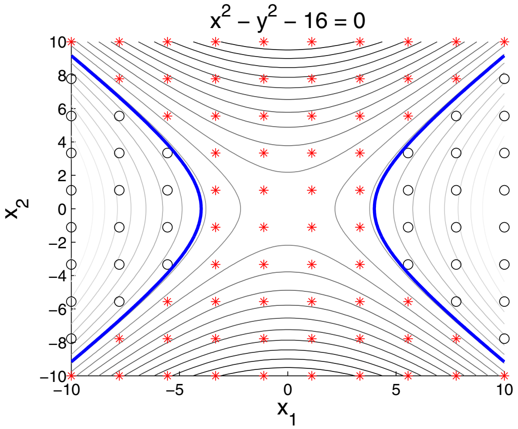


Fig. 5. Contours of the hyperbolic paraboloid which intersects with the Euclidean plane

Training by the PSO algorithm starts from the parameters (weights and bias) encoding .The individual particle structure is present in Figure 6. The parameter $\overline{a}_k$ describes the weight values between the connection of the input vector to the k neuron in the hidden layer. The parameter $\overline{w}_{i,j}$ describes the weight values

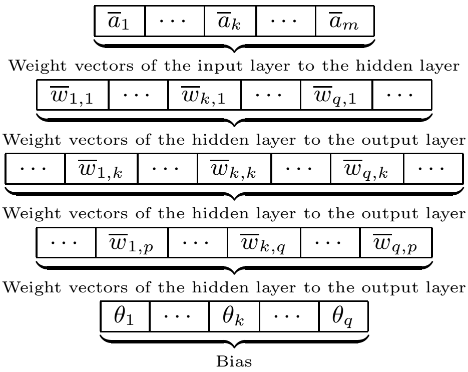

$\overline{a}_1$ · · · $\overline{a}_k$ · · · $\overline{a}_m$
Weight vectors of the input layer to the hidden layer
$\overline{w}_{1,1}$ · · · $\overline{w}_{k,1}$ · · · $\overline{w}_{q,1}$ · · ·
Weight vectors of the hidden layer to the output layer
· · · $\overline{w}_{1,k}$ · · · $\overline{w}_{k,k}$ · · · $\overline{w}_{q,k}$ · · ·
Weight vectors of the hidden layer to the output layer
· · · $\overline{w}_{1,p}$ · · · $\overline{w}_{k,q}$ · · · $\overline{w}_{q,p}$
Weight vectors of the hidden layer to the output layer
θ_1 · · · θ_k · · · θ_q
Bias

Fig. 6. Structure of a single particle of the PSO

between the connection of the ith neuron in the hidden layer and the jth neuron in the output layer. Also, the parameter θ_k describes the bias value of the kth neuron in the output layer.

Particles are evaluated by their Root Mean Squared Error (RMSE). RMSE is given in the following equation, where n is the number of patterns, q is the number of neurons in the output layer, y is the output of a single neuron j, t is the target for the single neuron j, and each input pattern is denoted by i.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{q}(t_{(i,j)} - y_{(i,j)})^2}{n}} \tag{34}$$

5 Experiments

5.1 Experiment 1

The goal of this experiment is to show the decision region generated by each hyper-conic neuron, and the final decision region of the whole network. Some synthetic data were defined ad-hoc to show the 2-D conic surfaces. We used four neurons to separate the pattern classes; class 1 is represented by circles, and class 2 by triangles. Fig 7 presents the conic decision surfaces that separate the pattern classes. The PSO algorithm found the following conic functions for the neurons: one neuron works with an ellipse, three neurons work with hyperbolas. Notice that the goal of this experiment is to show how the HC-MLP generates a complex decision region, however, during the training the generalization error was never measured (observe the unusual shape of the decision region).

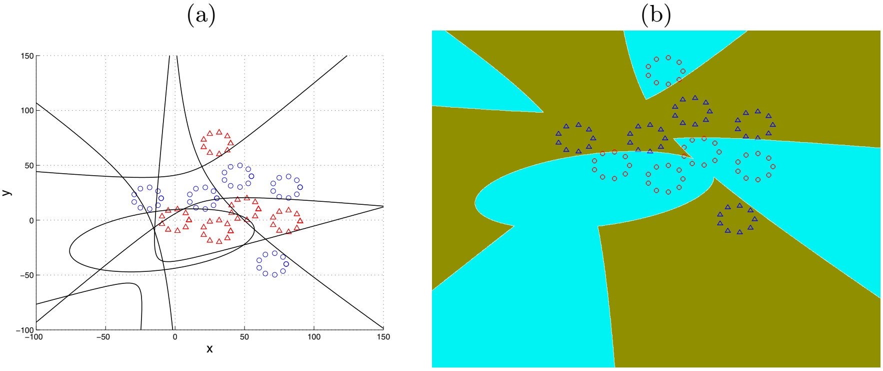


Fig. 7. (a) Conic surfaces generated by the neurons, (b) Final decision surface or region created by the HC-MLP to separate the two classes

5.2 Experiment 2

A HC-MLP and a MLP were trained and tested on the Fisher's Iris data set. The data set contains 3 classes, each class with 50 4-dimensional vectors. The classes refer to the type of iris plant, where one class is linearly separable from the other two, but some of these two are not linearly separable from the third class. A data set was divided in 45 training data set vectors and 105 test data set vectors as in [6]. The topology for both HC-MLP and MLP consisted of an input layer, a hidden layer and an output layer. The output layer indicates the outputs (0,0), (0,1), and (1,0), respectively, to indicate each of the three classes. The configuration of the number of neurons in the hidden layer varied between 2 up to 6 neurons. For training both models we used the described PSO algorithm. The PSO parameters were: inertia value 0.7298, 1.49618 for both social and cognitive coefficients, and 40 particles in the swarm. The stop criteria was the convergence to 1e-5 of the best particle, however, we used cross validation technique to prevent overtraining. PSO algorithm is a stochastic method, therefore 30 trials are carried out with the same parameters in order to provide statistical results. The statistical results of the training and test phase for the MLP and the HC-MLP are reported in Tables 1 and 2, respectively.

Table 1. Statistical Results for the MLP

		RMSE		**Clasification Rate**	
Neurons	**Epochs**	**Training**	**Testing**	**Training**	**Testing**
2	300	0.0425	0.4159	99.64 %	94.86 %
3	289	0.0473	0.4281	99.29 %	94.24 %
4	291	0.0369	0.4107	99.71 %	94.80 %
5	342	0.0548	0.4267	99.78 %	94.42 %
6	359	0.0426	0.4118	99.78 %	94.64 %

Table 2. Statistical Results for the HC-MLP

		RMSE		Clasification Rate	
Neurons	**Epochs**	**Training**	**Testing**	**Training**	**Testing**
2	300	0.0522	0.3603	99.41 %	97.05 %
3	289	0.0232	0.3608	99.56 %	96.55 %
4	292	0.0173	0.3987	99.56 %	95.94 %
5	342	0.0312	0.4123	99.64 %	94.99 %
6	359	0.0437	0.4083	99.56 %	95.09 %

Fig. 8 shows the average of RMSE and the average of the Classification Rate on testing data vectors for 30 runs.

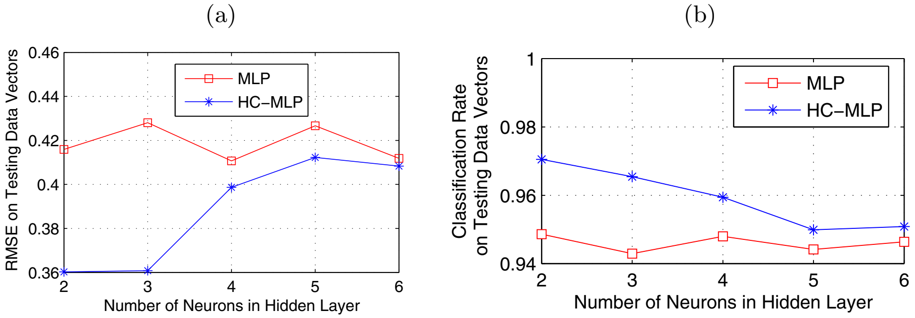


Fig. 8. (a) Statistical results for the RMSE (mean) on testing data vectors, (b) Statistical results for the classification rate (mean) on testing data vectors

6 Conclusions

In this paper a HC-MLP is presented. There are problems where a hyperplane is insufficient to generate complex decision surfaces that can separate non-linear separable classes. HC-MLP can be applied with high success to certain data types, particularly to classify compact regions. The algorithm chooses the best conic section that separates the data set vectors using circles, parabolas, hyperbola, and ellipses. The HC-MLP had a better generalization error than the MLP (with the same size). This means a significant advantage for the HC-MLP using a common data set. As noted, in compact data sets the HC-MLP will perform better. More experiments with data sets of a different nature are in progress. However, the main HC-MLP model will not change significantly in the future.

Acknowledgments. The authors gratefully acknowledge the financial support from National Council for Science and Technology of Mexico (CONACYT) and from the Center for Research in Mathematics (CIMAT).

References

1. Georgiou, G.M., Koutsougeras, C.: Complex Domain Backpropagation. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 39, 330–334 (1992)
2. Arena, P., Fortuna, L., Occhipinti, L., Xibilia, M.: Neural Networks for Quaternion-Valued Function Approximation. In: 1994 IEEE International Symposium on Circuits and Systems, ISCAS 1994, vol. 6, pp. 307–310 (1994)
3. Buchholz, S., Sommer, G.: A Hyperbolic Multilayer Perceptron. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, vol. 2, pp. 129–133 (2000)
4. Buchholz, S., Sommer, G.: On Clifford Neurons and Clifford Multi-Layer Perceptrons. Neural Networks 21, 925–935 (2008)
5. Bayro-Corrochano, Vallejo, E., Daniel, A.: Geometric Preprocessing, Geometric Feedforward Neural Networks and Clifford Support Vector Machines for Visual Learning. Neurocomputing 67, 54–105 (2005)
6. Perwass, C., Banarer, V., Sommer, G.: Spherical Decision Surfaces Using Conformal Modelling. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 9–16. Springer, Heidelberg (2003)
7. Perwass, C., Főrstner, W.: Uncertain Geometry with Circles, Spheres and Conics. In: Geometric Properties from Incomplete Data. Kluwer Academic Publishers (May 25, 2004)
8. Vince, J.: Geometric Algebra for Computer Graphics, 1st edn. Springer (April 14, 2008)
9. Engelbrecht Andries, P.: Fundamental of Computational Swarm Intelligence. Wiley (2005)

Vehicle Lateral Dynamics Fault Diagnosis Using an Autoassociative Neural Network and a Fuzzy System

Juan Pablo Nieto González and Pedro Pérez Villanueva

Corporación Mexicana de Investigación en Materiales,
COMIMSA S.A. de C.V.
Ciencia y Tecnología No. 790, Fracc. Saltillo 400, C.P. 25290
Saltillo, Coahuila, México
{juan.nieto,pperez}@comimsa.com

Abstract. The main goals of a fault diagnosis system in a vehicle are to prevent dangerous situations for occupants. This domain is a complex system that turns the monitoring task a very challenging one. On one hand, there is an inherent uncertainty caused by noisy sensor measurements and unmodeled dynamics and in the other hand the existence of false alarms that appears in a natural way due to the high correlation between several variables. The present work is a variant of a proposal made by the author. This paper presents a new approach based on history data process that can manage the variable correlation and can carry out a complete fault diagnosis system. In the first phase, the approach learns behavior from normal operation of the system using an autoassociative neural network. On a second phase a fuzzy system is carried out in order to diminish the presence of false alarms that could be originated by the noise presence and then a competitive neural network is used to give the final diagnosis. Results are shown for a ten variables vehicle monitoring.

Keywords: Vehicle Dynamics, Fault Detection, Fault Diagnosis, Autoassociative Neural Network, Competitive Neural Network, Fuzzy System, History Data.

1 Introduction

Engineering systems fault diagnosis is related to the detection of faults in complex machinery by detecting specific patterns of behavior in observed data. A modern vehicle is an example of such complex engineering system where there are a large number of sensors, controllers, and computer modules embedded in the vehicle that collect abundant signals. Vehicle fault diagnosis relies on the processing of such signals, which have dynamic ranges in magnitude, oscillation, frequency, slope, derivative, etc. Therefore, it is extremely difficult to develop a complete diagnostic using a model based method that can fully answer all questions related to automotive engineering faults. A vehicle fault detection and

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 236–246, 2013.

diagnosis using an autoassociative neural network (AANN) and a Fuzzy System (FS) is proposed. This system is a process history base method; only normal operating conditions data is needed for training the system. The organization of the paper is as follows. Section 2 reviews the state of the art. Section 3 gives the preliminaries and the background knowledge on AANN and FS. Section 4 gives the approach general description. Section 5 shows how this framework works in a simulation example. Finally, conclusion ends this paper in section 6.

2 State of the Art

Research related to automated or semiautomated highway vehicles has seen a tremendous amount of progress during the last years. Vehicle fault diagnosis relies on the processing of signals that most of the times include non-linear and noisy characteristics. In the past, different combinations of techniques have been done to deal with this problem, but in general we can find works that use model based approaches in order to carry out a complete fault diagnosis system. [1] describes a system for the detection of critical situations occurred when driving a car. A bank of observers combined with estimators is designed. Residuals are compared using the sequential test of Wald. [2] proposed a fault tolerant sensor system based on modeling the vehicle dynamics. A Kalman filter and two-track model are designed to create analytical signals replacing the faulty sensor. By reconfiguring the yaw rate, the longitudinal acceleration or the lateral acceleration can be estimated with small discrepancies. A fault tolerant sensor system is the goal of this research. [3] presents a model based approach doing parameter estimation and using parity equation for symptom generation then it applies fuzzy logic to detect small faults. [4] uses a multi-model-based methodology for estimation of some variables. Using these estimation, residuals are generated and manipulated with a fuzzy system. Specific patterns are used to detect and identify the location and type of sensors faults. Even though there are important results in the aforementioned projects, no one considers the correlation between variables in order to find the root cause of the faults. In an early project, a diagnosis system based on a vehicle model and soft computing techniques was proposed. [5] shows an approach divided in two phases. First, the vehicle model is used to obtain a set of residuals. Second, a vector is formed with the residual and then, using fuzzy logic, a comparison against their normal operating limits is made to look for those residuals standing out of limits. Finally, a competitive neural network classifies the output, indicating the faulty component in the vehicle. [6] uses an approach that combines a Takagi-Sugeno fuzzy model to describe the nonlinear two degrees of freedom vehicle motion and a bank of observers, using sliding mode design techniques to estimate the system state vector. [7] gives an approach that combines the linear-quadratic control method and the control Lyapunov function technique. Now a new vehicle fault detection and diagnosis proposal based on the history data using an AANN and a FS is proposed.

3 Preliminars

3.1 Auto-associative Neural Network AANN

[8] proposed an AANN, used as a Non-Linear Principal Components Analysis (NLPCA) method to identify and remove correlations among problem variables as an aid to dimensionality reduction, visualization, and exploratory data analysis. NLPCA operates by training a feed-forward neural network to perform the identity mapping, where the network inputs are reproduced at the output layer. The network contains an internal bottleneck layer (containing fewer nodes than input or output layers), which forces the network to develop a compact representation of the input data, and two additional hidden layers. Fig. 1 shows the architecture of this AANN with five layers.

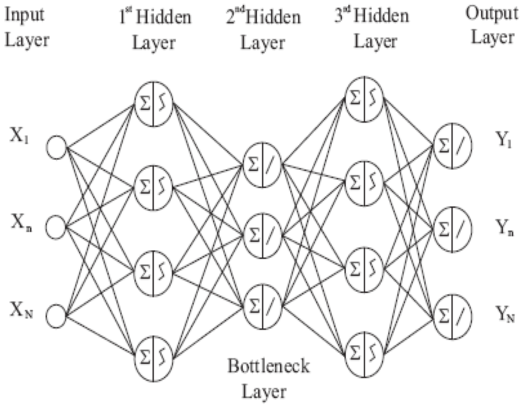


Fig. 1. Architecture of an AANN

This $AANN$ has one input layer and one output layer each with N neurons and three hidden layers with H_1, H_2 and H_3 neurons respectively. When an observation x is presented at the input of the network, the output neuron equation of the l^{th} layer is function of the neurons at the $(l-1)^{th}$ layer, given by equation 1, and shown in Fig. 2.

$$y_j = f\Big(\sum_{i=1}^{n(l-1)} w_{i,j}^{(l)} y_i^{(l-1)} \Big) \;\; l = 1, ..., 4 \tag{1}$$

where $y_j^{(0)} = x_n$, $n = 1, ..., N$ represents the components of the observations vector at the input of the network; $y_n^{(4)}$; $n = 1, ..., N$, represents the components of the estimation x given at the output of the network; $n(l-1)$ gives the number of neurons at the $(l-1)^{th}$ layer. Function $f(\dot{})$ is the neutron activation function,

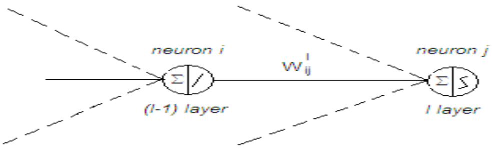


Fig. 2. Connection of the i^{th} neuron of the $(l-1)$ layer with the j^{th} neuron of the l layer

which is sigmoidal. The observations must be normalized as they lay inside of a unitary hypercube.

If the network has to learn a specific task, it is necessary to adjust the weights of the connections between neurons in order to minimize the difference between the expected and given output by the $AANN$. This minimization is carried out when computing the derivative of the error. The method commonly used is the back-propagation exploiting the gradient descendent method.

3.2 Fuzzy Logic

Fuzzy logic consists of the theory of fuzzy sets and possibility theory, and it was introduced by Zadeh in 1965 in order to represent and manipulate data that was not precise but rather fuzzy. Fuzzy Logic (FL) is a Soft Computing technique necessary for analyzing complex systems, especially where the data structure is characterized by several linguistic parameters. Fuzziness is a property of language. The human brain interprets imprecise and incomplete sensory information provided by perceptive organs. Fuzzy set theory provides a systematic calculus to deal with such information linguistically, and it performs numerical computation by using linguistic labels stipulated by membership functions.

The mathematical foundations of fuzzy logic rest in fuzzy set theory, which can be thought of as a generalization of classical set theory. In contrast to a classical set, a fuzzy set, is a set without a crisp boundary. That is, the transition from belong to a set to not belong to a set is gradual, and this smooth transition is characterized by membership functions, that give fuzzy set flexibility in modeling commonly used linguistic expressions.

Definitions and Terminology. Let X be a space of objects and x be a generic element of X. A classical set A, $A \subseteq X$, is defined as a collection of elements or objects $x \in X$, such that each x can either belong or not belong to the set A. By defining a characteristic function for each element x in X, it could be represented a classical set A by a set of ordered pairs $(x, 0)$ or $(x, 1)$, which indicates $x \notin A$ or $x \in A$, respectively.

Definition: *Fuzzy sets and membership functions*
If X is a collection of objects denoted generically by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)|x \in X\} \tag{2}$$

Where $\mu_A(x)$ in equation 2 is called the membership function (MF) for the fuzzy set A. The MF maps each element of X to a membership grade or membership value between 0 and 1. Then a fuzzy set is completely characterized by its MF. Usually X is referred to as the universe of discourse. Since most fuzzy sets have a universe of discourse X consisting of the real line R, it would be impractical to list all the pairs defining an MF. Thus, an MF should be expressed as a mathematical formula. Some typical examples of MF among others are: Triangular, Trapezoidal, Gaussian and Sigmoidal MF.

– **Linguistic variable** *is a variable whose values are words or sentences in a natural or synthetic language.* For example, the arguments for the linguistic variable *temperature* may be LOW, MEDIUM, and HIGH.
– **Fuzzy If-Then rule** *in which the antecedent and consequents are propositions containing linguistic variables.* They encode knowledge about a system in statements of the form:
  $If\ (a\ set\ of\ conditions\ are\ satisfied)$
  $Then\ (a\ set\ of\ consequences\ can\ be\ inferred)$
  Thus, a fuzzy If-Then rule assumes the form shown in expression 3:

$$If\ \ x\ \ is\ \ A\ \ Then\ \ y\ \ is\ \ B \tag{3}$$

  Where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y, respectively. Often $x\ \ is\ \ A$ is called the antecedent or premise, while $y\ \ is\ \ B$ is called the consequence or conclusion.

The starting point and the heart of a fuzzy system is a knowledge base consisting of the so called fuzzy If-Then rules as those showed above. Next step is to combine these rules into a single system.

4 Framework Description

According to [9] this proposal is a process history-based method because the need of a data set when the system runs under normal operating conditions. The general fault detection and diagnosis framework is shown in Fig. 3.

The difference between [5] and this work is the way the residuals are obtained. During the first phase of diagnosis, the former paper obtains the residuals when comparing the results of a vehicle model and the sensors measurements. In the present paper, the residuals are generated by the use of the AANN. First of all an standardization procedure is performed in order to manage all variables over a same scale. Then, a subset of 80% of the total data set is randomly generated. Using this subset the AANN model is learned. A five layers AANN is a network whose outputs are trained to emulate the inputs over an appropriate dynamic range. This characteristic is very useful to monitor variables of complex systems that have some degree of correlation with each other. Hence, each output receives

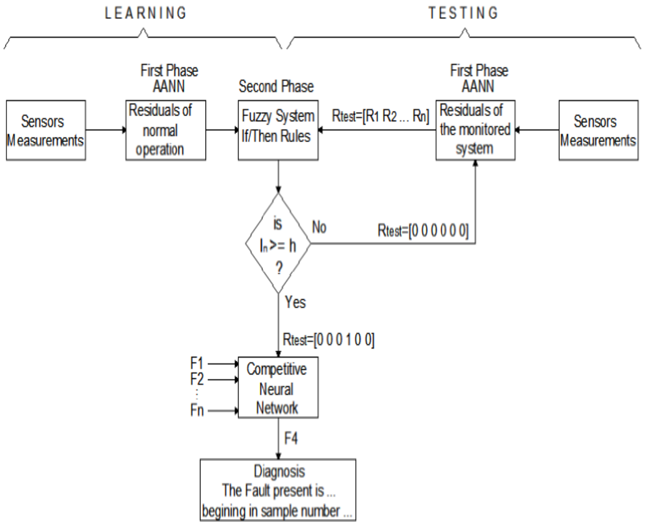


Fig. 3. General fault detection and diagnosis framework

some information from almost every input. During training, to make each output equal to the corresponding input, the interrelationships between all the input variables and each individual output are embedded in the connection weights of the network. As a result, any specific output, even the corresponding output shows only a small fraction of the input change over a reasonably large range. This allows the AANN to detect a failure by simply comparing each input with the corresponding output, obtaining in this way the residuals. After this, the limits for these residuals under normal operating conditions are computed. After that, a second phase based on fuzzy logic, looks for those residuals out of normal operation limits. Then, a competitive neural network classifies each output of the fuzzy system to know the actual operation mode of a system. For the second phase, firstly a data set of the residuals coming from the normal operation of the system is required. Then the minimum, maximum and statistical mean values for each of these residuals are obtained to design the rules of the fuzzy system. They will be used to verify if a residual is out of its normal operation limits or there is noise presence. In this work, we use the gaussian membership function for the fuzzy system. The set of fuzzy rules are stated in this form:

- If value is inside limits then R_n is zero
- If value is outside limits then R_n is one

Where R_n is the residual obtained from the comparison between the n^{th} output of the AANN and the measurement of the n^{th} sensor, in other words the n^{th} input of the AANN. The first phase delivers the residuals in a vector of real numbers R_{test} of dimension $1 \times n$:

$$R_{test} = [R_1, ..., R_n] \tag{4}$$

The R_{test} vector in expression 4, is analyzed by the fuzzy system to obtain the residuals out of limits. If this is the case, the fuzzy system assigns this residual a value of one, otherwise, is indicated with a zero. In order to detect false alarms, an index must be defined to monitor the number of consecutive points, given by the fuzzy system, that lay out of their given limits. In this way we can distinguish between noise presence or the existence of a real fault. Thus, this index I_n will count the consecutive number of times or points the residual R_n has out of limits, in such way that when I_n reaches h, that is a constant number set by the user, a binary R^b_{test} vector will be formed as that shown in equation ??. Here, it is shown for instance, the R^b_{test} formed when the fourth residual R_4 of an R_{test} vector formed by six residuals, has reached h consecutive points out of its limits.

$$R^b_{test} = [0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0] \tag{5}$$

This vector is the input of a competitive neural network classifier. For instance the vector shown in expression 5 could indicate a fault in the right rear wheel sensor. As this process is carried out over each of the R_n residuals forming the R_{test} vector, it is possible to know in which time or sample number a fault started.

4.1 Algorithm of the Proposal

The algorithm of the learning process for both phases can be summarized from step (1) to (8) as follows:

1. Take a normal operating conditions data set.
2. Randomly take a subset (80%) of the total amount of data.
3. Normalize the data subset. Train the AANN and learn the model.
4. Generate the normal operating conditions residuals
5. Obtain the minimum, maximum and mean values for each residual of the set of normal operation data from step (4).
6. Build the fuzzy system for each residual of the system being monitored according with the values obtained in step (5) with the $if then$ rules of the form:
   - If value is inside limits then R_n is zero
   - If value is outside limits then R_n is one
7. Define an index I_n for each residual in order to distinguish between noise presence and a fault presence. If $(I_n \geq h)$ Then a fault is present on R_n Else measurement noise is present on R_n. h is the threshold of consecutive points out of the limit allowed before setting an alarm of a possible fault presence, and it is set by the user.
8. Learn a competitive neural network with the different fault signatures in order to classify the output of the fuzzy system.

   When monitoring a process, the diagnostic system performs as follows:
9. Build a residual test vector of the form
   $R_{test} = [R_1, ..., R_n]$

10. Take the R_{test} vector as the input of the fuzzy system. The output of it is a R^b_{test} binary vector that will have a 0 if the residual is inside its normal operation limits and will have a 1 if it is out of them.
11. Verify the index of each residual I_n. If the points laying out of limits are consecutive Then increment the index I_n and Go to step (12) Else reset I_n to zero.
12. If $I_n \geq h$ Then stop the monitoring process and Go to step (13) Else return to step (9).
13. Set the R^b_{test} vector obtained in step (10) as the input of the competitive neural network previously trained. Give which fault is present and its location

5 Case Study

This proposal was tested with measurement data coming from VE-DYNA [1] (i.e. a vehicle simulator). Ten variables were monitored in this system, Table 1

Table 1. Description of variables

Variable	Description	Type
δ_f	Fron wheel steer angle	Input
CMar	Motor couple applied at rear wheels	Input
a_x	Acceleration in x-axis	Output
a_y	Acceleration in y-axis	Output
LFW	Left Front Wheel angular velocity	Output
RFW	Right Front Wheel angular velocity	Output
LRW	Left Rear Wheel angular velocity	Output
RRW	Right Rear Wheel angular velocity	Output
V_x	Longitudinal velocity gravity's center	Output
Yaw	Yaw velocity	Output

The diagnosis system was tailored according to the steps described on subsection 4.1. The experiments were performed by simulating a vehicle maneuver in a highway of double change of lane. A database, called chicane, was generated from VE-DYNA with 501 points, and conformed the normal operating conditions of system. Randomly 400 points (80% of data) were obtained of the database and formed a matrix whose dimension is 400 × 10. This matrix serves as the 400 examples for each of the 10 variables in the learning process of the *AANN*. The remaining data of normal operating conditions were exploited for validation. Fig. 4 shows the normal operating conditions space created by the output of the AANN.

[1] www.tesis.de

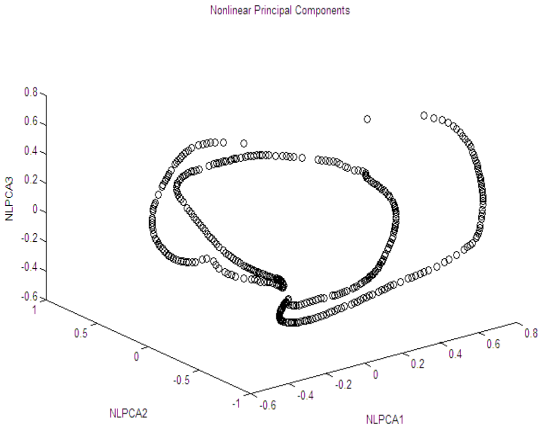


Fig. 4. Output of the AANN for the Normal operating conditions space defined by the AANN

Several simulations were carried out, where faults with different magnitudes on different number of samples were introduced at any time.

Faulty LFW. A fault of a change of 11% over the normal operating conditions in the left front wheel angular velocity (LFW) was introduced from sample number 45 to 57. Fig. 5 shows how the 3 non-linear principal components (NLPC) of the faulty data lie out of the space depicted by the 3 NLPC under normal operating conditions, indicating that a fault is present on the system. Note how it is easy to see the 13 samples that are in faulty mode, such that the FS just corroborates the fault presence.

After looking the output of the AANN the FS gives the residual vector identifying the variable that has a fault present on it. Fig. 6 shows the identification of a fault present by the FS.

For the present example, the FS gives the following residual vector

$$R_{test} = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ ] \tag{6}$$

which at a time is the input to the previously trained competitive neural network that gives the final diagnosis. In this case, according to the equation 6 and the classification made by the competitive neural network the final diagnosis indicates that there is a fault present on *LFW* variable as expected.

Table 2. Performance of detection for different magnitudes of measurement noise present in all the residuals simultaneously for different *h* values

Noise Magnitude	**h = 10**	**h = 7**	**h = 5**	**h = 3**
0.0001	100%	100%	95%	80%
0.001	100%	100%	85%	45%
0.01	100%	90%	65%	25%
0.1	100%	75%	25%	5%

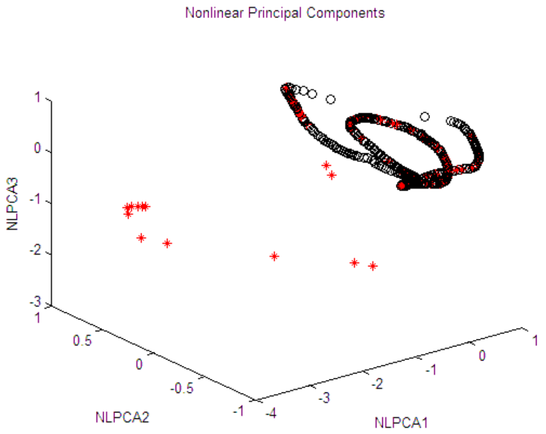


Fig. 5. Output of the AANN for chicane database when a fault of an increase 11% in 13 samples of *LFW* was induced.

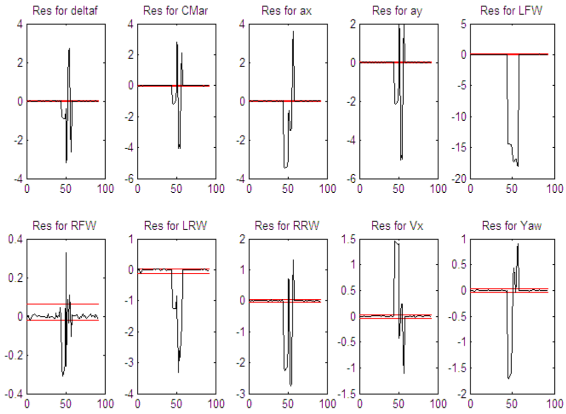


Fig. 6. Fuzzy system fault identification

Table 2 shows the results obtained after 50 different simulations scenarios for each row of it. Each simulation scenario considers different measurement noise magnitudes present in all the residuals simultaneously for each of the different quantities of consecutive points h. According with this table, the approach presents its better performance when $h = 10$ because with this index, the methodology could distinguish between noise presence and fault present along all the simulations carried out.

6 Conclusions

This paper has presented an online fault detection framework for a vehicle, based on the history data process. The diagnosis is carried out in two phases. The first phase uses a combination of an autoassociative neural network to obtain the residuals between the normal operation behavior and the information coming from sensors. The second phase is compose by a fuzzy system that could distinguish between noise or fault presence and a competitive neural network that classifies the output of the system in order to give the final diagnosis. As the system diagnoses each residual individually, this approach detects which variable is in faulty mode and its time of occurrence.

Acknowledgments. Authors thanks the support received by the CONACYT-FORDECYT project 143332 to do this research work.

References

1. Zbiri, N., Rabhi, A., Sirdi, N.K.M.: Analytical Redundancy Techniques for Faults in Vehicle. In: Mediterranean Conference on Control and Automation, Limassol, Cyprus, June 27-29, pp. 1585–1590 (2005)
2. Unger, I., Isermann, R.: Fault Tolerant Sensors for Vehicle Dynamics Control. In: American Control Conference, Minneapolis, USA, June 14-16 (2006)
3. Fisher, D., Borner, M., Isermann, R.: Fault detection for lateral and vertical vehicle dynamics. Control Engineering Practice 15(3), 315–324 (2007)
4. Quet, P.F., Salman, M.: Model-based Sensor Fault Detection and Isolation for X-By-Wire Vehicles Using a Fuzzy Logic System with Fixed Membership Functions. In: American Control Conference, New York, USA, pp. 2314–2319 (July 2007)
5. Nieto, J.P., Garza, L.E., Rabhi, A., El Hajjaji, A.: Fault Diagnosis of a Vehicle with Soft Computing Methods. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 492–502. Springer, Heidelberg (2008)
6. Oudghiri, M., Chadli, M., El Hajjaji, A.: Control and Sensor Fault-Tolerance of Vehicle Lateral Dynamics. In: Proceedings of the 17th World Congress. The International Federation of Automatic Control, Seoul, Korea, July 6-11, pp. 123–128 (2008)
7. Yang, H., Cocquempot, V., Jiang, B.: Optimal fault-tolerant path-tracking control for 4WS4WD electric vehicles. IEEE Transactions on Intelligent Transportation Systems 11(1), 237–243 (2010)
8. Kramer, M.: Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. AIChe Journal 37(2), 233–243 (1991)
9. Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N.: A Review of Process Fault Detection and Diagnosis Part I: Quantitative Model-Based Methods. Computers and Chemical Eng. 27(3), 293–311 (2003)
10. Simulator VE-DYNA, `http://www.tesis.de/index.php`

Modular Neural Networks Optimization with Hierarchical Genetic Algorithms with Fuzzy Response Integration for Pattern Recognition

Daniela Sánchez, Patricia Melin, Oscar Castillo, and Fevrier Valdez

Tijuana Institute of Technology
pmelin@tectijuana.mx

Abstract. In this paper a new model of a Modular Neural Network (MNN) with fuzzy integration based on granular computing is proposed. The topology and parameters of the MNN are optimized with a Hierarchical Genetic Algorithm (HGA). The proposed method can divide the data automatically into sub modules or granules, chooses the percentage of images and selects which images will be used for training. The responses of each sub module are combined using a fuzzy integrator, the number of the fuzzy integrators will depend of the number of sub modules or granules that the MNN has at a particular moment. The method was applied to the case of human recognition to illustrate its applicability with good results.

Keywords: Modular Neural Networks, Type-2 Fuzzy Logic, Hierarchical Genetic Algorithms, Granular computing.

1 Introduction

Hybrid intelligent systems are computational systems that integrate different intelligent techniques. This integration is mainly performed because each technique can have some individual limitations, but these limitations can be overcome if these techniques are combined.

These systems are now being used to support complex problem solving. These techniques are divided into two categories: traditional hard computing techniques and soft computing techniques (such as fuzzy logic, neural networks, and genetic algorithms).

There are many works where have been implemented this kind of systems, these works have combined different techniques and they have demonstrated that the integration of different intelligent techniques provide good results, such as in [19][20][21][22][24][25][29].

In this paper different intelligent techniques are combined such as neural networks, type-2 fuzzy logic and genetic algorithms using a granular approach. The pattern recognition based on biometrics is of great importance due to the fact that it can have a greater control about who has access to information or area, besides of allowing identifying if a person is the one who claims to be, for this reason the proposed method was applied to pattern recognition based on ear biometrics.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 247–258, 2013.

This paper is organized as follows: Section 2 contains the basic concepts used in this research work. Section 3 contains the general architecture of the proposed method. Section 4 presents experimental results and in Section 5, the conclusions of this work are presented.

2 Basic Concepts

In this section we present a brief overview of the basic concepts used in this research work.

2.1 Modular Neural Network

Neural networks (NNs) can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques [2][16].

A computational system can be considered to have a modular architecture if it can be divided into two or more subsystems in which each individual subsystem evaluates either distinct inputs or the same inputs without communicating with the other subsystems. The concept of modularity is an extension of the principle of divide and conquer, the problem should be divided into smaller sub problems that are solved by experts (modules) and their partial solutions should be integrated to produce a final solution [2][30]. The results of the different applications involving Modular Neural Networks (MNNs) lead to the general evidence that the use of modular neural networks implies a significant learning improvement comparatively to a single NN and especially to the backpropagation NN. Each neural network works independently in its own domain. Each of the neural networks is build and trained for a specific task [1][18].

2.2 Granular Computing

The term “Granular Computing (GrC)” was first introduced in 1997 by T.Y. Lin to label a new field of multi-disciplinary study [17][39][42]. Zadeh [41] first introduced the notion of information granulation in 1979 and suggested that fuzzy set theory may find potential applications in this respect, which pioneers the explicit study of granular computing. With the concept of his information granulation, Zadeh further presented granular mathematics [43]. The consideration of granularity is motivated by the practical needs for simplification, clarity, low cost, approximation, and tolerance of uncertainty [10][34][35][36][37].

Granular computing has begun to play important roles in different areas [3][39][40]. A granule may be interpreted as one of the numerous small particles forming a larger unit. At least three basic properties of granules are needed: internal properties reflecting the interaction of elements inside a granule, external properties revealing its interaction with other granules and, contextual properties showing the relative existence of a granule in a particular environment [38].

2.3 Type-2 Fuzzy Logic

Fuzzy logic is an area of soft computing that enables a computer system to reason with uncertainty [4]. The concept of a type-2 fuzzy set was introduced by Zadeh (1975) as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0,1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. When we cannot determine the membership of an element in a set as 0 or 1, fuzzy sets of type-1 are used. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], fuzzy sets of type-2 are used [12][13][14]. Uncertainty in the primary memberships of a type-2 fuzzy set, Ã, consists of a bounded region that we call the "footprint of uncertainty" (FOU). Mathematically, it is the union of all primary membership functions [7][6][23]. The basics of fuzzy logic do not change from type-1 to type-2 fuzzy sets, and in general will not change for type-n. A higher type number just indicates a higher degree of fuzziness [5].

2.4 Hierarchical Genetic Algorithms

A Genetic Algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection where the fittest individuals survive [28]. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" [11][26]. Genetic Algorithms (GAs) are nondeterministic methods that employ crossover and mutation operators for deriving offspring. GAs work by maintaining a constant-sized population of candidate solutions known as individuals (chromosomes) [8][15][27].

Introduced in [31], a Hierarchical genetic algorithm (HGA) is a type of genetic algorithm. Its structure is more flexible than the conventional GA. The basic idea under hierarchical genetic algorithm is that for some complex systems, which cannot be easily represented, this type of GA can be a better choice. The complicated chromosomes may provide a good new way to solve the problem [32][33].

3 General Architecture of the Proposed Method

The proposed method combines modular neural networks (MNN) and fuzzy logic in each module as response integrators. This proposed method can use a dataset that will be divided into different numbers of sub modules, and each sub module is divided into other 3 sub modules. The general architecture of the proposed method is shown in Fig. 1. This method can be used for some applications such as time series prediction or human recognition, in this work the method is used for human recognition based on ear biometric. This method was developed in MATLAB 2012a and using Type-2 Fuzzy Logic Toolbox developed by [7] for responses integration.

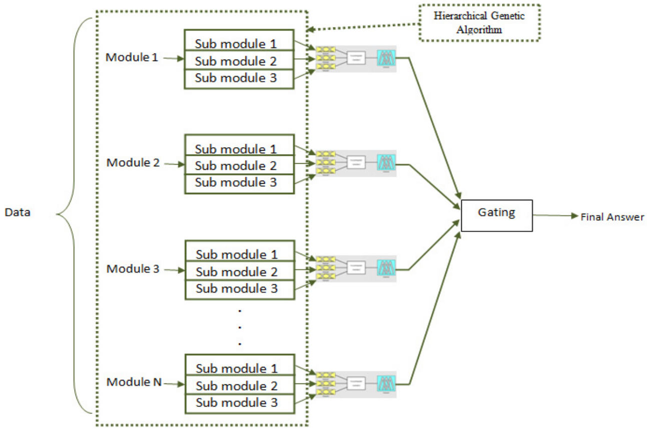


Fig. 1. The general architecture of proposed method

The proposed method for MNN consists in changing the number of modules and the data per module, for example in the case of human recognition; it means there will be different number of persons in each sub module. In this moment the granulation is sequential, perhaps in the future can be established in another way. The responses of each sub module are combined using type-2 fuzzy logic and finally a gating is used. In Fig. 2 the fuzzy integrator used is shown. This fuzzy integrator has 3 inputs because each images is divided into 3 parts, this pre-processing is described below.

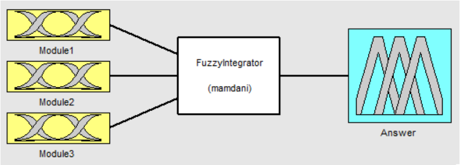


Fig. 2. Fuzzy integrator used

An example of the memberships functions used in each input and the output is shown in Fig. 3.

It is important to say that this method also chooses randomly which images will be used for training, but first the percentage of images for training is established (this percentage can be defined randomly or using a genetic algorithm).The proposed method also performs the optimization of the modular neural networks (as number of layers, goal error, number of neurons, etc.). The hierarchical genetic algorithm developed for this optimization is described below.

3.1 Description of the Hierarchical Genetic Algorithm for MNN Optimization

The proposed hierarchical genetic algorithm allows the optimization of the number of modules and the percentage of data for training, besides of some parameters of the

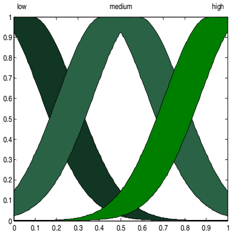


Fig. 3. Example of the membership functions

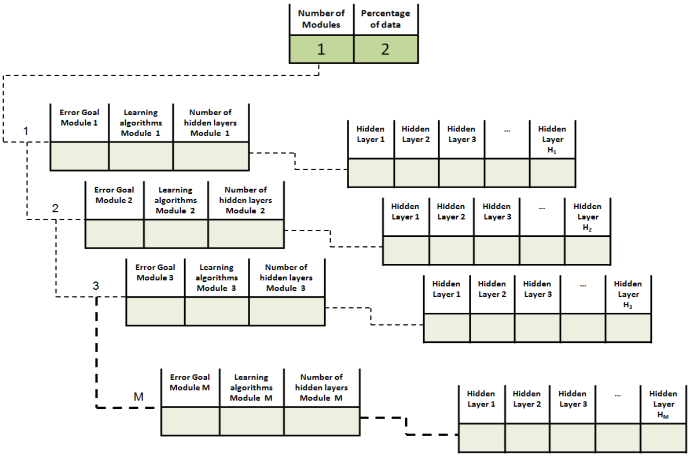


Fig. 4. The chromosome of hierarchical genetic algorithm for the MNN

modular neural networks such as the number of hidden layers, number of neurons per hidden layer, error goal and learning algorithm per module. The chromosome of the proposed hierarchical genetic algorithm is shown in Fig. 4.

The fitness function is described below:

$$f = \sum\nolimits_{i=1}^{m} \left(\left(\sum_{j=1}^{n_m} X_j \right) \Big/ n_m \right) \tag{1}$$

where $\boldsymbol{m}$ is the total number of modules, $\boldsymbol{X_j}$ is 0 if the module provides the correct result and 1 if not, and $\boldsymbol{n_{\mathrm{m}}}$ is total number of data points used for testing in the corresponding module. The main objective of this HGA is to minimize the error recognition.

3.2 Database of Ear

The database selected for the tests performed in this work is of the University of Science and Technology of Beijing (USTB) [9]. The database consists of 77 people, which contains 4 images per person (one ear), the image dimensions are 300 x 400 pixels, and the format is BMP. An example of pre-processing applied to each image of the ear is shown in Fig. 5. A manual cut is performed to remove the parts of the image that are not in our interest, then make a re-size of images to 132 x 91 pixels and finally the image is automatically divided in the 3 parts of interest (helix, lobe and shell).

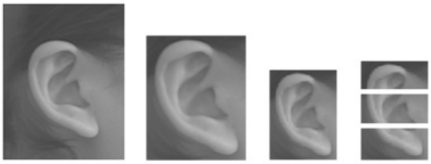


Fig. 5. Sample pre-processing done to the images of ear

It means that each sub module is divided into other 3 sub modules, due to the fact that each image is divided in 3 parts that is independently of the number of modules or granules indicated by our method.

4 Experimental Results

This section describes the tests of the proposed method and we focused on human recognition. In this section the results obtained in this work are presented.

4.1 Results with 3 Modules

In this test the number of modules was established in 3 and the database was divided equally among the 3 modules, the variables that were established randomly were the percentage of images used and the images which would be used for training, also the number of neurons of two hidden layers and the learning algorithm. For the integration of responses in each module, type-2 fuzzy logic was used in this and the following tests. In Table 1 the best results for ear with 3 modules are shown. In the second column the images that were used for training are shown. The best recognition rate obtained in this test was 98.70% using 3 images for training.

4.2 Results with Different Number of Modules

In this test the number of modules is randomly and the database was also divided randomly among the different modules, the variables that remain established randomly were the percentage of images used and the images which would be used for training, also the number of neurons of two hidden layers and the learning algorithm. In Table 2 the best results for ear in this test are shown.

Table 1. The best results for ear (with 3 modules)

Training	Images for training	Persons per module	Neurons	Recognition Rate
1	12% (4)	Module # 1 (1 to 26) Module # 2 (27 to 52) Module # 3 (53 to 77)	216,300 228,282 188,250	56.27% (130/231)
2	74% (2,3 and 4)	Module # 1 (1 to 26) Module # 2 (27 to 52) Module # 3 (53 to 77)	184,234 205,239 252,297	98.70% (76/77)
3	55% (1 and 4)	Module # 1 (1 to 26) Module # 2 (27 to 52) Module # 3 (53 to 77)	71,191 79,291 247,90	55.19% (85/154)
4	43% (1 and 2)	Module # 1 (1 to 26) Module # 2 (27 to 52) Module # 3 (53 to 77)	249,162 103,310 210,232	75.97% (117/154)
5	45% (2 and 4)	Module # 1 (1 to 26) Module # 2 (27 to 52) Module # 3 (53 to 77)	77,137 108,90 65,334	59.74% (92/154)

Table 2. The best results for the ear (with different number of modules)

Training	Images for training	Persons per module	Neurons	Recognition Rate
1	66% (2,3 and 4)	Module # 1 (1 to 61) Module # 2 (62 to 77)	126,241 197,272	94.80% (73/77)
2	73% (1, 2 and 4)	Module # 1 (1 to 8) Module # 2 (9 to 13) Module # 3 (14 to 29) Module # 4 (30 to 39) Module # 5 (40 to 49) Module # 6 (50 to 51) Module # 7 (52 to 58) Module # 8 (59 to 74) Module # 9 (75 to 77)	260,74 119,112 79,340 96,281 291,271 120,229 127,273 277,270 107,52	80.51% (62/77)
3	78% (2,3 and 4)	Module # 1 (1 to 7) Module # 2 (8 to 57) Module # 3 (58 to 77)	265,141 124,124 255,119	97.40% (75/77)
4	53% (3 and 4)	Module # 1 (1 to 23) Module # 2 (24 to 30) Module # 3 (31 to 49) Module # 4 (50 to 63) Module # 5 (64 to 75) Module # 6 (76 to 77)	94,90 280,167 296,267 229,71 86,117 197,235	81.16% (125/154)
5	49% (2 and 4)	Module # 1 (1 to 2) Module # 2 (3 to 5) Module # 3 (6 to 20) Module # 4 (21 to 38) Module # 5 (39 to 58) Module # 6 (59 to 61) Module # 7 (62 to 67) Module # 8 (68 to 74) Module # 9 (75 to 77)	144,223 107,91 106,98 68,59 229,247 96,106 267,77 197,281 233,55	77.27% (119/154)

4.3 Optimized Results

The proposed hierarchical genetic algorithm was used in this test. In this test some parameters were optimized with the finality of obtaining optimal parameters for number of modules, percentage of data for training, goal error, learning algorithm, number of hidden layers and their number of neurons for each module.

4.3.1 Parameter of Percentage of Data for Training until 80%

It was established that the hierarchical genetic algorithm could work with a percentage of data for training until 80%, the results obtained were shown in Table 3.

Table 3. The best results for ear (parameter of percentage of data for training until 80%)

	Images for training	Persons per module	Num. Hidden layers and Num. of neurons	Rec. Rate	Duration
1	71% (2,3 and 4)	Module # 1 (1 to 7) Module # 2 (8 to 20) Module # 3 (21 to 32) Module # 4 (33 to 40) Module # 5 (41 to 44) Module # 6 (45 to 58) Module # 7 (59 to 61) Module # 8 (62 to 77)	2 (169,179) 4 (97,180,56,48) 4 (29,42,114,173) 4 (116,74,66,21) 1 (192) 2 (133,42) 3 (69,162,172) 3 (77,72,125)	100% (77/77)	02:54:51
2	75% (1,2 and 3)	Module # 1 (1 to 14) Module # 2 (15 to 30) Module # 3 (31 to 40) Module # 4 (41 to 57) Module # 5 (58 to 73) Module # 6 (74 to 77)	3 (30,161,36) 5 (65,130,174,159,99) 1 (155) 3 (35,119,71) 1 (101) 1 (33)	100% (77/77)	03:43:07
3	78% (2,3 and 4)	Module # 1 (1 to 15) Module # 2 (16 to 30)	5 (170,23,29,140,179) 4 (54,131,167,128)	97.40% (75/77)	06:15:45

In this test in almost all the evolutions a rate of recognition of 100% is obtained, this when 3 images were used for training. These results were obtained before of the maximum number of generations. The graph of the convergence of the evolution # 1 is shown in Fig. 6.

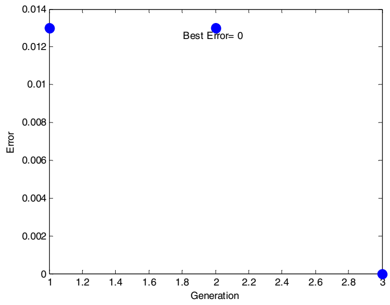


Fig. 6. Convergence evolution #1

4.3.2 Parameter of Percentage of Data for Training until 50%

When the genetic algorithm uses 3 images a recognition rate of 100% is obtained, for this reason in this test the hierarchical genetic algorithm could work with a percentage of data for training until 50%, because a better recognition rate using less images for training is wanted to have.

Table 4. The best results for ear (parameter of percentage of data for training until 50%)

	Images for training	Persons per module	Num. Hidden layers and Num. of neurons	Rec. Rate	Duration
4	41% (2 and 3)	Module # 1 (1 to 4) Module # 2 (5 to 24) Module # 3 (25 to 44) Module # 4 (45 to 53) Module # 5 (54 to 66) Module #6 (67 to 77)	5(95,56,190,43,197) 3 (61,82,76) 2 (115,141) 4 (118,41,173,77) 1 (133) 2 (141,48)	93.50% (144/154)	58:13:56
5	48% (2 and 3)	Module # 1 (1 to 7) Module # 2 (8 to 22) Module # 3 (23 to 32) Module # 4 (33 to 34) Module # 5 (35 to 56) Module # 6 (57 to 77)	5 (31,50,68,146,54) 5 (55,168,67,145,79) 1 (48) 2 (36,137) 2 (195,175) 2 (99,56)	94.80% (146/154)	41:42:55
6	45% (2 and 3)	Module # 1 (1 to 4) Module # 2 (5 to 21) Module # 3 (22 to 30) Module # 4 (31 to 40) Module # 5 (41 to 56) Module # 6 (57 to 66) Module # 7 (67 to 77)	4 (148,64,150,117) 4 (167,116,120,160) 4 (129,131,130,139) 1 (74) 2 (159,126) 3 (42,22,30) 3 (162,172,66)	94.15% (145/154)	45:30:17

In this test the best recognition rate obtained was 94.80% using 2 images in the training phase in the training #5. The graph of the convergence of the evolution # 5 is shown in Fig. 7.

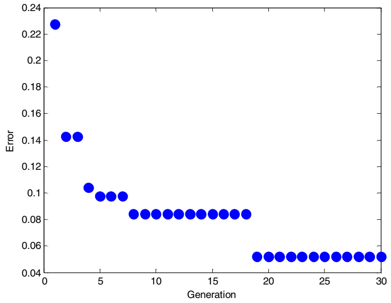


Fig. 7. Convergence evolution #5

5 Conclusions

A new method of modular neural networks with type-2 fuzzy logic integration using a granular approach was proposed. The main goals of the proposed method were: to know the number of modules, data per module, and percentage of data for training, all that with finality of obtaining a better rate of recognition.

A hierarchical genetic algorithm was developed for optimization of some parameters of this model of modular neural networks; those parameters are number of modules, percentage of data for training, error goal per module, number of hidden layers per module and their respective neurons.

This method was applied for pattern recognition based on the ear biometrics, and better results were obtained when the proposed hierarchical genetic algorithm is used, but these results can be improved if a genetic algorithm for the fuzzy integrators is developed. This method can be used with any biometric measure, no matter number of persons or number of images per person, due to this method can be adapted easily.

References

1. Auda, G., Kamel, M.S.: Modular Neural Networks A Survey. Int. J. Neural Syst. 9(2), 129–151 (1999)
2. Azamm, F.: Biologically Inspired Modular Neural Networks, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (2000)
3. Bargiela, A., Pedrycz, W.: The roots of granular computing. In: IEEE International Conference on Granular Computing (GrC), pp. 806–809 (2006)
4. Castillo, O., Melin, P.: Soft Computing for Control of Non-Linear Dynamical Systems. Springer, Heidelberg (2001)
5. Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications, pp. 29–43. Springer, Berlin (2008)
6. Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A.: Building Fuzzy Inference Systems with a New Interval Type-2 Fuzzy Logic Toolbox. Transactions on Computational Science 1, 104–114 (2008)
7. Castro, J.R., Castillo, O., Melin, P.: An Interval Type-2 Fuzzy Logic Toolbox for Control Applications. In: FUZZ-IEEE 2007, pp. 1–6 (2007)
8. Coley, A.: An Introduction to Genetic Algorithms for Scientists and Engineers, Wspc, Har/Dskt edn. (1999)
9. Database Ear Recognition Laboratory from the University of Science & Technology Beijing (USTB). Found on the Web page, `http://www.ustb.edu.cn/resb/en/index.htmasp` (accessed September 21, 2009)
10. Han, J., Dong, J.: Perspectives of Granular Computing in Software Engineering. In: GrC 2007, pp. 66–71 (2007)
11. Haupt, R., Haupt, S.: Practical Genetic Algorithms, 2nd edn., pp. 42–43. Wiley-Interscience (2004)
12. Hidalgo, D., Castillo, O., Melin, P.: Optimization with genetic algorithms of modular neural networks using interval type-2 fuzzy logic for response integration: The case of multimodal biometry. In: IJCNN 2008, pp. 738–745 (2008)

13. Hidalgo, D., Castillo, O., Melin, P.: Type-1 and Type-2 Fuzzy Inference Systems as Integration Methods in Modular Neural Networks for Multimodal Biometry and Its Optimization with Genetic Algorithms. In: Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Hybrid Intelligent Systems. SCI, vol. 154, pp. 89–114. Springer, Heidelberg (2008)
14. Hidalgo, D., Melin, P., Licea, G., Castillo, O.: Optimization of Type-2 Fuzzy Integration in Modular Neural Networks Using an Evolutionary Method with Applications in Multimodal Biometry. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS, vol. 5845, pp. 454–465. Springer, Heidelberg (2009)
15. Huang, J., Wechsler, H.: Eye Location Using Genetic Algorithm. In: Second International Conferenceon Audio and Video-Based Biometric Person Authentication, pp. 130–135 (1999)
16. Khan, A., Bandopadhyaya, T., Sharma, S.: Classification of Stocks Using Self Organizing Map. International Journal of Soft Computing Applications 4, 19–24 (2009)
17. Lin, T.Y.: Granular Computing. Announcement of the BISC Special Interest Group on Granular Computing (1997)
18. Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: An Evolutionary Approach for Neural Networks and Fuzzy Systems, 1st edn., pp. 119–122. Springer (2005)
19. Melin, P., Kacprzyk, J., Pedrycz, W.: Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition. Springer (2009)
20. Melin, P., Mendoza, O., Castillo, O.: An improved method for edge detection based on interval type-2 fuzzy logic. Expert Syst. Appl. 37(12), 8527–8535 (2010)
21. Melin, P., Mendoza, O., Castillo, O.: Face Recognition with an Improved Interval Type-2 Fuzzy Logic Sugeno Integral and Modular Neural Networks. IEEE Transactions on Systems, Man, and Cybernetics, Part A 41(5), 1001–1012 (2011)
22. Melin, P., Sánchez, D., Castillo, O.: Genetic optimization of modular neural networks with fuzzy response integration for human recognition. Information Sciences 197, 1–19 (2012)
23. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
24. Mendoza, O., Melin, P., Castillo, O.: Interval type-2 fuzzy logic and modular neural networks for face recognition applications. Appl. Soft Comput. 9(4), 1377–1387 (2009)
25. Mendoza, O., Melin, P., Licea, G.: A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral. Inf. Sci. 179(13), 2078–2101 (2009)
26. Mitchell, M.: An Introduction to Genetic Algorithms, 3rd edn. A Bradford Book (1998)
27. Nawa, N., Takeshi, F., Hashiyama, T., Uchikawa, Y.: "A study on the discovery of relevant fuzzy rules using pseudobacterial genetic algorithm. IEEE Transactions on Industrial Electronics 46(6), 1080–1089 (1999)
28. Raikova, R.T., Aladjov, H.Ts.: Hierarchical genetic algorithm versus static optimization investigation of elbow flexion and extension movements. Journal of Biomechanics 35, 1123–1135 (2002)
29. Sánchez, D., Melin, P.: Modular Neural Network with Fuzzy Integration and Its Optimization Using Genetic Algorithms for Human Recognition Based on Iris, Ear and Voice Biometrics. In: Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Recognition Based on Biometrics. SCI, vol. 312, pp. 85–102. Springer, Heidelberg (2010)
30. Santos, J.M., Alexandre, L.A., Marques de Sá, J.: Modular Neural Network Task Decomposition Via Entropic Clustering. In: ISDA (1), pp. 62–67 (2006)

31. Tang, K.S., Man, K.F., Kwong, S., Liu, Z.F.: Minimal Fuzzy Memberships and Rule Using Hierarchical Genetic Algorithms. IEEE Trans. Ind. Electron. 45(1), 162–169 (1998)
32. Wang, C., Soh, Y.C., Wang, H., Wang, H.: A Hierarchical Genetic Algorithm for Path Planning in a Static Environment with Obstacles. In: Canadian Conference on Electrical and Computer Engineering, IEEE CCECE 2002, vol. 3, pp. 1652–1657 (2002)
33. Worapradya, K., Pratishthananda, S.: Fuzzy supervisory PI controller using hierarchical genetic algorithms. In: 5th Asian Control Conference, vol. 3, pp. 1523–1528 (2004)
34. Yao, J.T.: A ten-year review of granular computing. In: Proceedings of the 3rd IEEE International Conference on Granular Computing (GrC), pp. 734–739 (2007)
35. Yao, J.T.: Information granulation and granular relationships. In: Proceedings of 2005 IEEE Conference on Granular Computing (GrC), pp. 326–329 (2005)
36. Yao, Y.: A Partition Model of Granular Computing. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Swiniarski, R., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 232–253. Springer, Heidelberg (2004)
37. Yao, Y.Y.: Granular computing: basic issues and possible solutions. In: Proceedings of the 5th Joint Conferences on Information Sciences, pp. 186–189 (2000)
38. Yao, Y.Y.: On Modeling Data Mining with Granular Computing. In: 25th International Computer Software and Applications Conference (COMPSAC), pp. 638–649 (2001)
39. Yao, Y.Y.: Perspectives of granular computing. In: IEEE International Conference on granular computing (GrC), pp. 85–90 (2005)
40. Yu, F., Pedrycz, W.: The design of fuzzy information granules: Tradeoffs between specificity and experimental evidence. Applied Soft Computing 9(1), 264–273 (2009)
41. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R. (eds.) Advances in Fuzzy Set Theory amd Applications, pp. 3–18. North-Holland Publishing Co. (1979)
42. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Computing 2, 23–25 (1998)
43. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 19, 111–127 (1997)

Neural Network with Type-2 Fuzzy Weights Adjustment for Pattern Recognition of the Human Iris Biometrics

Fernando Gaxiola, Patricia Melin, Fevrier Valdez, and Oscar Castillo

Tijuana Institute of Technology, Tijuana México
pmelin@tectijuana.mx

Abstract. In this paper a neural network learning method with type-2 fuzzy weight adjustment is proposed. The mathematical analysis of the proposed learning method architecture and the adaptation of type-2 fuzzy weights are presented. The proposed method is based on research of recent methods that handle weight adaptation and especially the use of fuzzy weights. In this work an ensemble neural network of three neural networks and the use of average integration to obtain the final result is presented. The proposed approach is applied to a case of time series prediction to illustrate the advantage of using type-2 fuzzy weights.

Keywords: Type-2 Fuzzy Logic, Neural Network, Iris Biometric, Pattern Recognition, Fuzzy Weights.

1 Introduction

This paper is focused on the area of neural networks with type-2 fuzzy weights for pattern recognition based on biometric measures, specifically in the recognition by the human iris biometric measurement. At present, biometric measurements are being widely used for person recognition systems. A lot has been said about the use of such measures, particularly for the signature, fingerprint, face and voice. As more research was done in this area further biometric measures were discovered, among which the human iris by its peculiarity of not losing over the years its universality and authenticity.

In order to get a good identification, we propose two neural networks with type-2 fuzzy weights; two fuzzy inference systems are used to obtain the type-2 fuzzy weights, one is used in the hidden layer and the other in the output layer for both neural networks. The input for the two neural networks is the database of human iris, and some methods or techniques are used for pre-processing the images such as normalization, resizing, cut, and edge detection, among several others.

The rest of the paper is organized as follows: Section 2 contains a brief explanation from previous research about adaptations of weights in neural networks and methods of recognition of persons with human iris, and basic concepts relevant to the area, section 3 presented the description of problem addressed in this paper, section 4 defines the method proposed for this research, and in section 5 presents the results achieved in this work and Section 6 draws conclusions and future work.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 259–270, 2013.

2 Background and Basic Concepts

2.1 Neural Network

An artificial neural network (ANN) is a distributed computing scheme based on the structure of the nervous system of humans. The architecture of a neural network is formed by connecting multiple elementary processors, this being an adaptive system that has an algorithm to adjust their weights (free parameters) to achieve the performance requirements of the problem based on representative samples [1] [2].

The most important property of artificial neural networks is their ability to learn from a training set of patterns, i.e. able to find a model that fits the data [3][4][5].

The artificial neuron consists of several parts (see Fig. 1). On one side are the inputs, weights, the summation, and finally the adapter function. The input values are multiplied by the weights and added: $\sum x_i w_{ij}$. This function is completed with the addition of a threshold amount i. This threshold has the same effect as an entry with value -1. It serves so that the sum can be shifted left or right of the origin. After addition, we have the function f applied to the sum, resulting the final value of the output, also called y_i [5], obtaining the following equation.

$$y_i = \left(\sum_{i=1}^{n} x_i w_{ij} \right) \qquad (1).$$

Where f may be a nonlinear function with binary output + -1, a linear function f (z) = z, or as sigmoidal logistic function:

$$f(z) = \frac{1}{1+e^{-z}}. \qquad (2).$$

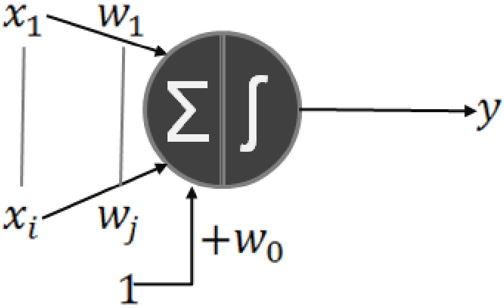


Fig. 1. Scheme of an artificial neuron

2.2 Fuzzy Logic

Fuzzy logic is an area of soft computing that enables a computer system to reason with uncertainty [6]. A fuzzy inference system consists of a set of if-then rules defined over fuzzy sets. Fuzzy sets generalize the concept of a traditional set by allowing the membership degree to be any value between 0 and 1 [7]. This corresponds, in the real world, to many situations where it is difficult to decide in an unambiguous manner if something belongs or not to a specific class [8]. The basic structure of a fuzzy inference system consists of three conceptual components: a rule base, which contains a selection of fuzzy rules, a database (or dictionary) which defines the membership functions used in the rules, and a reasoning mechanism that performs the inference procedure [9][10][11].

2.3 Type-2 Fuzzy Logic

The concept of a type-2 fuzzy set was introduced by Zadeh (1975) as an extension of the concept of an ordinary fuzzy set (henceforth called a "type-1 fuzzy set"). A type-2 fuzzy set is characterized by a fuzzy membership function, i.e., the membership grade for each element of this set is a fuzzy set in [0, 1], unlike a type-1 set where the membership grade is a crisp number in [0,1]. Such sets can be used in situations where there is uncertainty about the membership grades themselves, e.g., uncertainty in the shape of the membership function or in some of its parameters. Consider the transition from ordinary sets to fuzzy sets. When we cannot determine the membership of an element in a set as 0 or 1, we use fuzzy sets of type-1. Similarly, when the situation is so fuzzy that we have trouble determining the membership grade even as a crisp number in [0, 1], we use fuzzy sets of type-2 [12][13][14][15][16][17].

2.4 Historical Development

The most interest and important research in the area of recognition of persons with iris human for this work is presented here:

Daugman presented a new algorithm for the recognition of people using the biometric measurement of Iris [18].

Roy proposed an iris recognition system for the identification of persons using support vector machine [19].

Cho and Kim presented a new method to determine the winner in LVQ neural network [20].

Sarhan used the discrete cosine transform for feature extraction and artificial neural network for recognition [21].

Abiyev and Altunkaya presented the iris recognition system using neural network [22].

In the area of the backpropagation learning method exists many works about adjustment or managing of weights but only the most important and relevant for this research will be considered here [23][24][25][26][27][28][29][30][31][32][33][34][35]:

Kamarthi and Pittner [36], used extrapolation t to obtain a weight prediction of the network at a future epoch.

Ishibuchi et al. [37], used a fuzzy neural network where the weights are given as trapezoidal fuzzy numbers.

Ishibuchi et al. [38], proposed a fuzzy neural network architecture with symmetrical fuzzy triangular numbers for the fuzzy weights and biases, denoted by the lower, middle and upper limit of the fuzzy triangular numbers.

Feuring [39], developed a learning algorithm in which the backpropagation algorithm is used to compute the new lower and upper limits media weights. The modal value of the new fuzzy weight is calculated as the average of the new computed limits.

Castro et al. [40], uses interval type-2 fuzzy neurons for the antecedents and interval of type-1 fuzzy neurons for the consequents of the rules. This approach handles the weights as numerical values to determine the input of the fuzzy neurons, as the scalar product of the weights for the input vector.

3 Problem Description

This work focuses primarily on the development of the method of the neural network with type-2 fuzzy weights and the identification of individuals. This problem is considered in many works in this area, considering various measures to achieve it with biometrics (fingerprint, voice, palm of hand, signature) and various methods of identification (with particular emphasis on neural networks) [41].

We used a database of human Iris from the Institute of Automation of the Chinese Academy of Sciences (CASIA) (see Fig. 2). It consists of 9 images per person, for a total of 10 individuals, giving a total of 90 images. The image dimensions are 320 x 280, JPEG format [42][43][44].

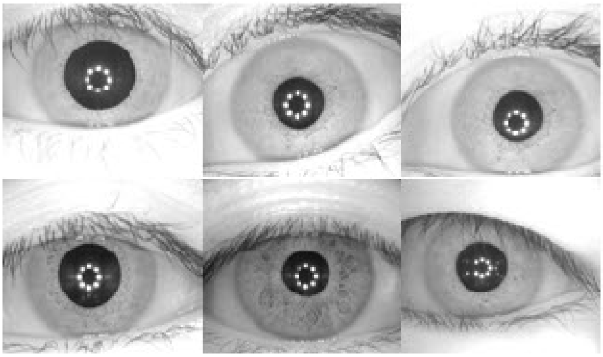


Fig. 2. Examples of the human iris images from the CASIA database

4 Proposed Neural Network with type-2 Fuzzy Weights

The work was focused on the recognition of persons using a modular neural network with type-2 fuzzy weights, and image preprocessing methods to obtain the interest region of the iris.

4.1 Iris Image Pre-processing

The images of the human iris that are introduced to the two neural networks were preprocessed as follows:

- Obtain the coordinates and radius of the iris and pupil [45].
- Making the cut in the Iris.
- Resize the cut of the Iris to 21-21 pixels.
- Convert images from vector to matrix
- Normalize the images.

4.2 Neural Network with Type-2 Fuzzy Weights

The proposed architecture with two neural networks with type-2 fuzzy weights consist of 120 neurons in the hidden layer and 10 neurons in the output layer, the inputs are the preprocessed iris images with a total of 10 persons (60 for training – 60 for test in total) (see Fig. 3). The inputs vary in ±5 percent between the two networks.

We considered two neural networks for managing type-2 fuzzy weights in each hidden layer and output layer. In each hidden layer and output layer we used a type-2 fuzzy inference system to obtain the new weights in each epoch of the network.

The two neural networks used the learning method that updates weight and bias values according to the resilient backpropagation algorithm. The updates of the weights are adapted to manage type-2 fuzzy weights.

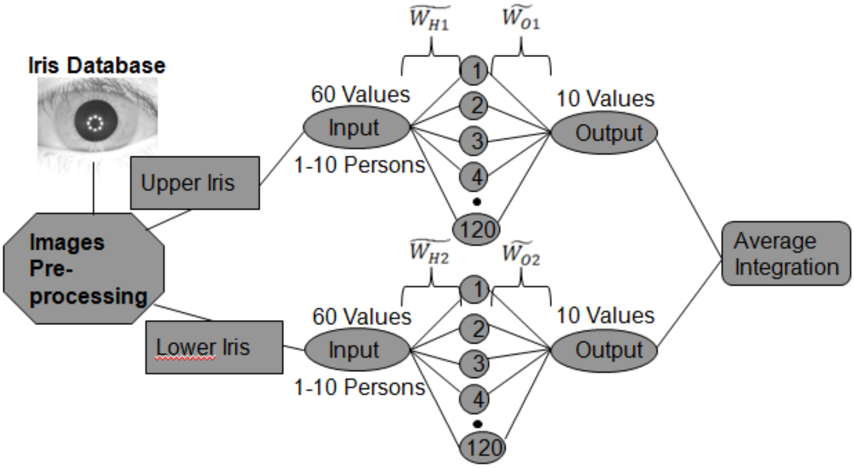


Fig. 3. Proposed neural networks with type-2 fuzzy weights architecture

We used four type-2 fuzzy inference systems to obtain the new weights, one for the hidden layer in the one network and the second network, and one for the output layer in the first network and the second network. The four type-2 fuzzy inference system consists of two inputs (actual weight and change of weight) and one output (new weight) (see Fig. 4); the two inputs have two triangular membership functions, and the output have two triangular membership functions: for the hidden layer in the first network (see Fig. 5), for the output layer in the first network (see Fig. 6), for the hidden layer in the second network (see Fig. 7), and for the output layer in the second network (see Fig. 8); and work with six rules (see Fig. 9).

The integration of the two networks is realized with average integration.

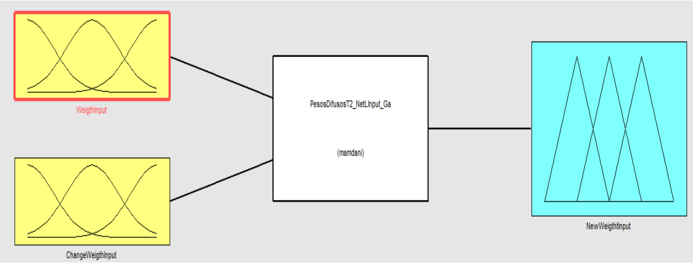
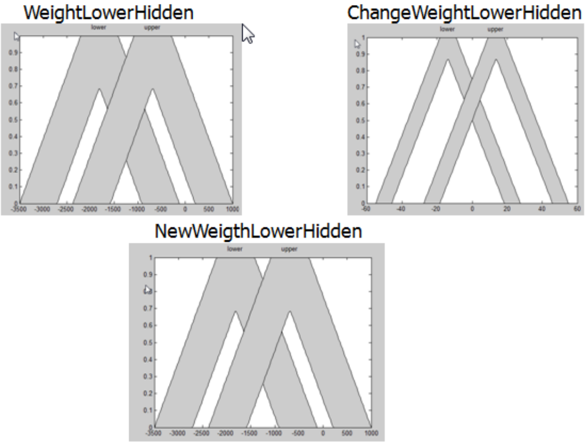


Fig. 4. Structure for the four type-2 fuzzy integration system


WeightLowerHidden

ChangeWeightLowerHidden

NewWeigthLowerHidden

Fig. 5. Inputs and Output for the type-2 fuzzy inference system for the hidden layer in the first network

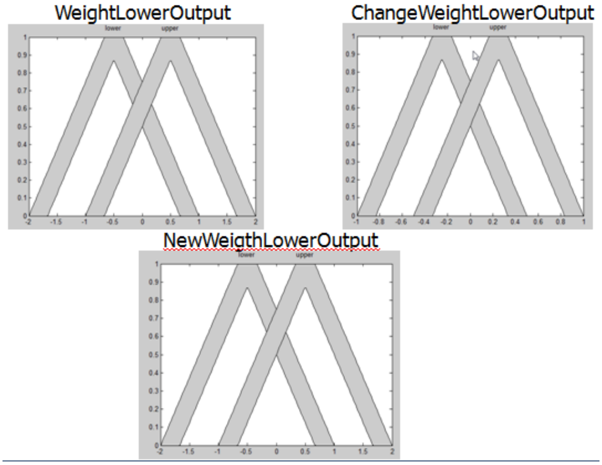


Fig. 6. Inputs and Output for the type-2 fuzzy inference system for the output layer in the first network

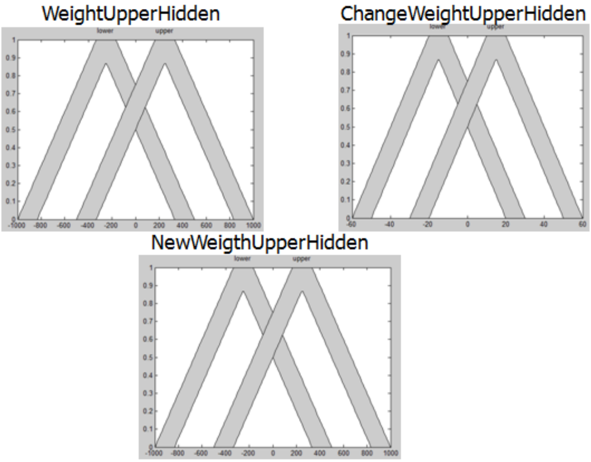


Fig. 7. Inputs and Output for the type-2 fuzzy inference system for the hidden layer in the second network

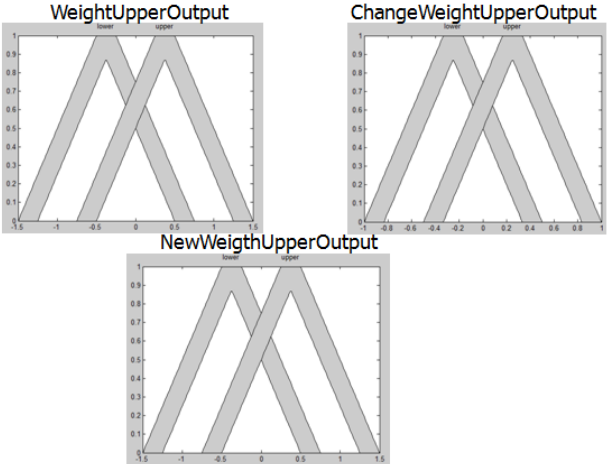
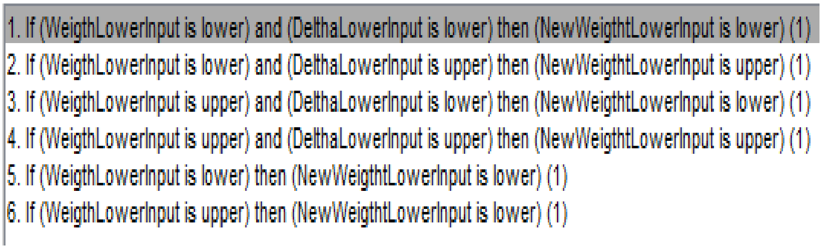


Fig. 8. Inputs and Output for the type-2 fuzzy inference system for the output layer in the second network



1. If (WeigthLowerInput is lower) and (DelthaLowerInput is lower) then (NewWeigthtLowerInput is lower) (1)
2. If (WeigthLowerInput is lower) and (DelthaLowerInput is upper) then (NewWeigthtLowerInput is upper) (1)
3. If (WeigthLowerInput is upper) and (DelthaLowerInput is lower) then (NewWeigthtLowerInput is lower) (1)
4. If (WeigthLowerInput is upper) and (DelthaLowerInput is upper) then (NewWeigthtLowerInput is upper) (1)
5. If (WeigthLowerInput is lower) then (NewWeigthtLowerInput is lower) (1)
6. If (WeigthLowerInput is upper) then (NewWeigthtLowerInput is lower) (1)

Fig. 9. Rules for the four type-2 fuzzy inference system

5 Simulation Results

Five tests were performed with the proposed modular neural network under the same conditions and the same database of the iris; in Table 1 we show the obtained results:

Table 1. Table of results from the experiments

No.	Epoch	Error	Time	Total Rec.
T1	12	0.01	72 min.	76.66 % (23/30)
T2	12	0.01	72 min.	70 % (21/30)
T3	12	0.01	72 min.	73.33 % (22/30)
T4	12	0.01	72 min.	83.33 % (25/30)
T5	12	0.01	71 min.	70 % (21/30)

The best result is a total recognition of 25 out of 30 images of iris of 10 persons; giving a recognition rate of 83.33 %.

The average of the 5 tests is 74.66 percent of recognition.

In Table 2, we present a comparison of results with those of other researchers obtained for iris recognition of persons, realized with different or similar methods.

Table 2. Table of comparison of results

Method.	Percentage of recognition
Proposed Method	83.33 %
Gaxiola [32]	97.13 %
Sanchez-Avila	97.89 %
Ma	98.00 %
Tisse	89.37 %
Daugman	99.90 %
Sarhan	96.00 %

6 Conclusions

In this paper we presented a two neural network architecture with type-2 fuzzy weights, which has as input the database of human iris images, with average integration. In this work, several methods were used to make the elimination of noise that the original pictures had until the coordinates of the center and radius were obtained, and then make a cut around the iris.

The type-2 fuzzy weights provide favorable results for recognition of a neural network, because the results obtained are on average of 74.66 percent of recognition and a best result of 83.33 percent of recognition, these results is good considering that the training method according to the resilient backpropagation algorithm is not good working with images.

The type-2 fuzzy weights manage the uncertainty allowing working with more complex patterns, which allows obtaining better results with noise in the image.

References

[1] Cazorla, M., Escolano, F.: Two Bayesian Methods for Junction Detection. IEEE Transaction on Image Processing 12(3), 317–327 (2003)

[2] Martinez, G., Melin, P., Bravo, D., Gonzalez, F., Gonzalez, M.: Modular Neural Networks and Fuzzy Sugeno Integral for Face and Fingerprint Recognition. In: Abraham, A., de Baets, B., Köppen, M., Nickolay, B. (eds.) Advances in Soft Computing. AISC, vol. 34, pp. 603–618. Springer, Heidelberg (2006)

[3] De Wilde, P.: The Magnitude of the Diagonal Elements in Neural Networks. Neural Networks 10(3), 499–504 (1997)

[4] Salazar, P.A., Melin, P., Castillo, O.: A New Biometric Recognition Technique Based on Hand Geometry and Voice Using Neural Networks and Fuzzy Logic. In: Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Hybrid Intelligent Systems. SCI, vol. 154, pp. 171–186. Springer, Heidelberg (2008)

[5] Phansalkar, V.V., Sastrq, P.S.: Analysis of the Back-Propagation Algorithm with Momentum. IEEE Transactions on Neural Networks 5(3), 505–506 (1994)

[6] Castillo, O., Melin, P.: Soft Computing for Control of Non-Linear Dynamical Systems. Springer, Heidelberg (2001)

[7] Zadeh, L.A.: Fuzzy Sets. Journal of Information and Control 8, 338–353 (1965)

[8] Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing, pp. 2–3. Springer, Heidelberg (2005)

[9] Okamura, M., Kikuchi, H., Yager, R., Nakanishi, S.: Character diagnosis of fuzzy systems by genetic algorithm and fuzzy inference. In: Proceedings of the Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications, Halong Bay, Vietnam, pp. 468–473 (1998)

[10] Wang, W., Bridges, S.: Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules. Department of Computer Science Mississippi State University, March 2 (2000)

[11] Jang, J., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice Hall, New Jersey (1997)

[12] Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications, pp. 29–43. Springer, Berlin (2008)

[13] Castro, J.R., Castillo, O., Melin, P.: An Interval Type-2 Fuzzy Logic Toolbox for Control Applications. In: FUZZ-IEEE, pp. 1–6 (2007)

[14] Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A.: Building Fuzzy Inference Systems with a New Interval Type-2 Fuzzy Logic Toolbox. Transactions on Computational Science 1, 104–114 (2008)

[15] Hidalgo, D., Castillo, O., Melin, P.: Type-1 and Type-2 Fuzzy Inference Systems as Integration Methods in Modular Neural Networks for Multimodal Biometry and Its Optimization with Genetic Algorithms. In: Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Hybrid Intelligent Systems. SCI, vol. 154, pp. 89–114. Springer, Heidelberg (2008)

[16] Sanchez, D., Melin, P.: Optimization of modular neural networks and type-2 fuzzy integrators using hierarchical genetic algorithms for human recognition. In: IFSA 2011, OS-414. Surabaya-Bali, Indonesia (2011)

[17] Sepúlveda, R., Castillo, O., Melin, P., Rodriguez, A., Montiel, O.: Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic. Information Sciences 177(11), 2023–2048 (2007)

[18] Daugman, J.: Statistical Richness of Visual Phase Information: Update on Recognizing Persons by Iris Patterns. International Journal of Computer Vision 45(1), 25–38 (2001)

[19] Roy, K., Bhattacharya, P.: Iris Recognition with Support Vector Machines. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 486–492. Springer, Heidelberg (2005)

[20] Cho, S., Kim, J.: Iris Recognition Using LVQ Neural Network. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 26–33. Springer, Heidelberg (2006)

[21] Sarhan, A.: Iris Recognition using Discrete Cosine Transform and Artificial Neural Networks. Journal of Computer Science 5, 369–373 (2009)

[22] Abiyev, R., Altunkaya, K.: Neural Network based Biometric Personal Identification with fast iris segmentation. International Journal of Control, Automation and Systems 7(1), 17–23 (2009)

[23] Barbounis, T.G., Theocharis, J.B.: Locally Recurrent Neural Networks for Wind Speed Prediction using Spatial Correlation. Information Sciences 177(24), 5775–5797 (2007)

[24] Gedeon, T.: Additive Neural Networks and Periodic Patterns. Neural Networks 12(4-5), 617–626 (1999)

[25] Meltser, M., Shoham, M., Manevitz, L.: Approximating Functions by Neural Networks: A Constructive Solution in the Uniform Norm. Neural Networks 9(6), 965–978 (1996)

[26] Yeung, D., Chan, P., Ng, W.: Radial Basis Function Network Learning using Localized Generalization Error Bound. Information Sciences 179(19), 3199–3217 (2009)

[27] Casasent, D., Natarajan, S.: A Classifier Neural Net with Complex-Valued Weights and Square-Law Nonlinearities. Neural Networks 8(6), 989–998 (1995)

[28] Draghici, S.: On the Capabilities of Neural Networks using Limited Precision Weights. Neural Networks 15(3), 395–414 (2002)

[29] Neville, R.S., Eldridge, S.: Transformations of Sigma–Pi Nets: Obtaining Reflected Functions by Reflecting Weight Matrices. Neural Networks 15(3), 375–393 (2002)

[30] Yam, J., Chow, T.: A Weight Initialization Method for Improving Training Speed in Feedforward Neural Network. Neurocomputing 30(1-4), 219–232 (2000)

[31] Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design, p. 736. PWS Publishing, Boston (1996)

[32] Fletcher, R., Reeves, C.M.: Function Minimization by Conjugate Gradients. Computer Journal 7, 149–154 (1964)

[33] Powell, M.J.D.: Restart Procedures for the Conjugate Gradient Method. Mathematical Programming 12, 241–254 (1977)

[34] Beale, E.M.L.: A Derivation of Conjugate Gradients. In: Lootsma, F.A. (ed.) Numerical Methods for Nonlinear Optimization, pp. 39–43. Academic Press, London (1972)

[35] Moller, M.F.: A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Networks 6, 525–533 (1993)

[36] Kamarthi, S., Pittner, S.: Accelerating Neural Network Training using Weight Extrapolations. Neural Networks 12(9), 1285–1299 (1999)

[37] Ishibuchi, H., Morioka, K., Tanaka, H.: A Fuzzy Neural Network with Trapezoid Fuzzy Weights, Fuzzy Systems. In: IEEE World Congress on Computational Intelligence, vol. 1, pp. 228–233 (1994)

[38] Ishibuchi, H., Tanaka, H., Okada, H.: Fuzzy Neural Networks with Fuzzy Weights and Fuzzy Biases. In: IEEE International Conference on Neural Networks, vol. 3, pp. 1650–165 (1993)

[39] Feuring, T.: Learning in Fuzzy Neural Networks. In: IEEE International Conference on Neural Networks, vol. 2, pp. 1061–1066 (1996)

[40] Castro, J., Castillo, O., Melin, P., Rodríguez-Díaz, A.: A Hybrid Learning Algorithm for a Class of Interval Type-2 Fuzzy Neural Networks. Information Sciences 179(13), 2175–2193 (2009)

[41] Sánchez, O., González, J.: Access Control Based on Iris Recognition. Technological University Corporation of Bolívar, Faculty of Electrical Engineering, Electronics and Mechatronics, Cartagena of Indias, Monography, pp. 1–137 (November 2003)

[42] Muron, A., Pospisil, J.: The human iris structure and its usages, Czech Republic, Physica, pp. 89–95 (2000)

[43] Ma, L., Wang, Y., Tan, T.: Iris recognition based on multichannel Gabor filtering. In: 5th Asian Conference on Computer Vision, ACCV 2002, Melbourne, Australia, vol. 1, pp. 279–283 (2002)

[44] Database of Human Iris. Institute of Automation of Chinese Academy of Sciences (CASIA). Found on the Web page, http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp

[45] Masek, L., Kovesi, P.: MATLAB Source Code for a Biometric Identification System Based on Iris Patterns. The School of Computer Science and Software Engineering, The University of Western Australia (2003)

[46] Gaxiola, F., Melin, P., López, M.: Modular Neural Networks for Person Recognition using the Contour Segmentation of the Human Iris Biometric Measurement. In: Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft-Computing for Recognition based on Biometrics. SCI, vol. 312, pp. 137–153. Springer, Heidelberg (2010)

Model Reference Adaptive Position Controller with Smith Predictor for a Shaking-Table in Two Axes

Carlos Esparza, Rafael Núñez, and Fabio González

Unidades Tecnológicas de Santander – UTS, Research Group on Advanced Control – GICAV,
Bucaramanga, Santander, Colombia
{carlosesfra,ing.rafaeln,fagonzarm}@gmail.com

Abstract. In structural behavior, the analysis of civil buildings by seismic tests has been generalized by the use of shaking tables. This method requires advanced control systems. In our research we show the implementation of a Model Reference Adaptive Control (MRAC) to control the position of a shaking table, modified by the introduction of a Smith predictor to compensate the error produced by the system delay. The mechanic is based on a Slider-crank device. The control system is implemented on a 32 bits platform by Microchip, and the control is done via a remote server using the RENATA network. The results of our adaptive control system were experimentally verified using the shaking table with a 24 kg mass as a load.

Keywords: Adaptive Control, MRAC, Smith Predictor, Shaking-Table, RENATA.

1 Introduction

Traditional control systems are sufficient to control any process in which it is knows the mathematical model of the plant, especially when the process is lineal. However, when the process model is unknown and it has non-linear components, it is necessary to implement other kind of control systems designs. These control systems should be independent from the model to control [1] [2]. One example of this problem is presented in this work, in which it is necessary to control a shaking table that is used to analyze the earthquake evaluations for tall buildings or several active antiseismic systems that decrease the oscillations of buildings subject to seismic events. Generally, shaking tables consists of electrohydraulic servo systems, hydraulic actuators, a table and sensor [3–6], but in the present work the shaking table used for the test has a different mechanism that generates the displacement on each axis table.

In this particular case, the shaking table is based on a slider-crank mechanism that uses an AC induction motor to generate the linear displacement through a mechanical system based in pulleys and with chains to articulate each axis. The motors are controlled by frequency inverters through the analog option by controlling a digital potentiometer. Figure 1 shows a blocks diagram that explains the entire system interconnection. Considering these factors, to obtain a mathematical model of the system that includes the mechanical and electrical components will be complex and with high uncertainty due to parameters that can't be measured. For that reason, a

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 271–282, 2013.

model for each axis was obtained using identification techniques. Both models have approximations higher than 90%, and they include the delay introduced by the pulley mechanism. These models are presented in section 2.

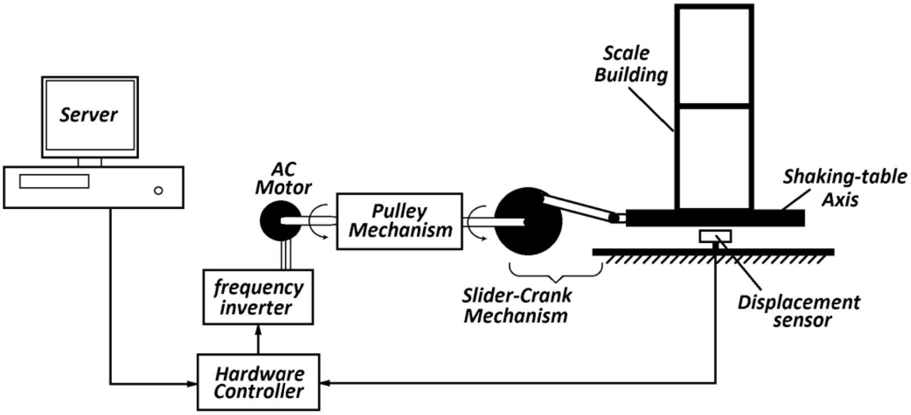


Fig. 1. Shaking table system interconnection

Although the implementation of an adaptive controller does not require a mathematical model to control the system, in the Model Reference Adaptive Control (MRAC) technique it is necessary to use a plant that simulates the ideal response of the system, which will be the signal that the controller try to follows to minimize the error [7]. The structure of the control system applied to our work is shown in figure 2.

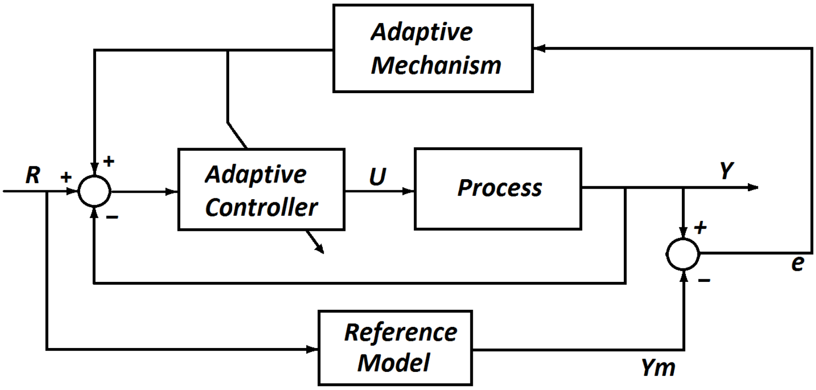


Fig. 2. Block diagram of MRAC Adaptive Model

A MRAC adaptive control is based on the MIT rule, so called because it was developed at Massachusetts Institute of Technology, that implements the gradient optimization technique to minimize the loss function given by equation 1, where *e* is the signal error in figure 1 and θ the controller parameters.

$$J(\theta) = \frac{1}{2}e^2 \ . \tag{1}$$

To minimize this function, the θ parameters must change in the direction of the negative gradient of J [8] [9] given by the equation 2, in which γ parameter is the adaptation gain, that must be correctly selected in order to reduce the settling time, and at the same time not rising the oscillation frequencies that cause an increase in the overshoot [10, 11].

$$\frac{d\theta}{dt} = -\gamma\frac{\partial J}{\partial\theta} = -\gamma e\frac{\partial e}{\partial\theta} = -\gamma e\left(\frac{\partial Y}{\partial\theta} - \frac{\partial Ym}{\partial\theta}\right). \tag{2}$$

The partial derivate of the error respect to the controller parameters (third term in equation 2) is the sensitivity derivative of the system. It explains how the error is affected by the adjustable controller parameters [8]. The equation 2 is the general form, and when there are several θ adjustable parameters, θ will be a vector and the partial derivate becomes in the gradient of error respect to controller parameters.

In this work, a position adaptive controller with two adaptation parameters θ (θ_1 and θ_2) is implemented, so that the control law is described as in equation 3 [12] [9]. Also, in this work the plant model is described by a second order plant with an integrator for the X-axis and a delay, and for simplification the delay is eliminated from the closed-loop plant model, to be added after the loop, in the same way as it is used in the Smith predictor structure. For the Y axis, the model is described in a similar way, but with a first order plant with an integrator. This is a good approximation of the model, because the changes on perturbations dynamics in the system are slow [13]. The Smith predictor structure used in this work is based in the Intern Model Control (IMC) technique as showed in figure 3.

$$U = \theta_i^T . w = \theta_1 R - \theta_2 Y \ . \tag{3}$$

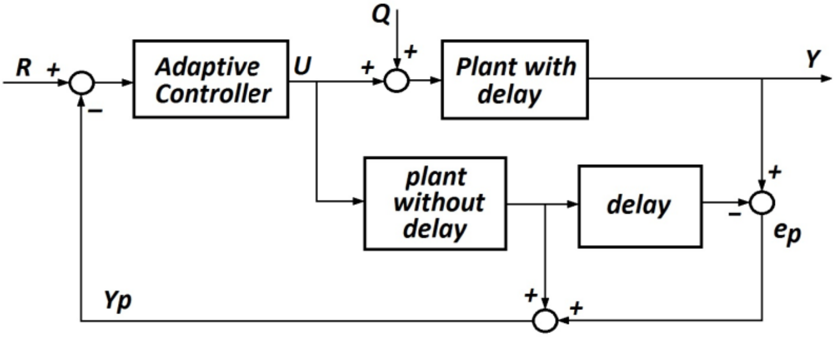


Fig. 3. Smith predictor control structure, IMC representation

2 Plant Models

2.1 X and Y Axes Models. Identification

The X axis (North-South component) and the Y axis (East-West component) of the shaking table are mechanical structures based on a slider-crank mechanism like it was described above, as it is shown on figure 4. To simplify the model of each axis, a linear model for displacement was obtained by the identification of a process model with transfer functions as shown in equation 4 for the X-axis and equation 5 for the Y-axis.

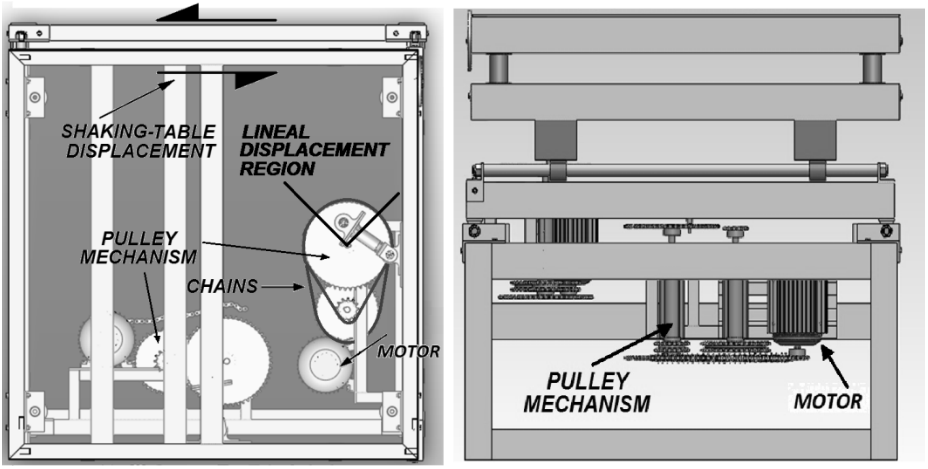


Fig. 4. Left: Slider-Crank mechanism used in the shaking table axis. Right: pulley mechanism.

$$G(s) = \frac{K_p . e^{-T_d * s}}{s(1 + 2.Zeta.Tw.s + (Tw.s)^2)} \; . \tag{4}$$

$$G(s) = \frac{K_p . e^{-T_d * s}}{s(1 + Tp1.s)} \; . \tag{5}$$

This lineal approximation can be made by defining the lineal displacement region of each table axis, as it is shown in figure 4, which allows a maximum displacement on each axis in the range of + 6 cm to -6 cm. In table 1 the parameters of equations 4 and 5 for X and Y axis are presented. They achieve approximations of 91.8% and 91.97% respectively. The final transfer functions are shown in equations 6 and 7, which are the plant models used for the Smith predictor in the branch without the delay.

Table 1. Model Parameters, obtained by identification process

Parameters	X axis	Y axis
K_P	82.739	77.977
T_d	0.075121	0.076994
Zeta	0.40121	--
Tw	0.029128	--
Tp1	--	0.021358

$$G_X(s) = \frac{82.739.e^{-0.075121s}}{s(0.0008484s^2 + 0.02337s + 1)} . \tag{6}$$

$$G_Y(s) = \frac{77.977.e^{-0.0077277s}}{s(0.02136\,s + \ 1)} . \tag{7}$$

Equations 6 and 7 are also used to obtain the reference model transfer function in the adaptive controller, and the transfer functions used in the loops that modify the control parameters θ.

2.2 Reference Models for X and Y Axes

From figure 2, the output of the process is the multiplication of the transfer functions of the plant $G_P(s)$ with the output of the controller $U(s)$ in Laplace domain. Given that the control output is defined by equation 3, and the plant model with delay can be approximated to a plant model without delay as it proposed in [13], the response Y for X axis can be described by equation 8.

$$Y_X(s) = \frac{K_p}{s(1 + 2.Zeta.Tw.s + (Tw.s)^2)}(\theta_1 R - \theta_2 Y_X) . \tag{8}$$

By solving the equation 8, the closed-loop transfer function that relates the output Y_X with the input R of the system is:

$$\frac{Y_X(s)}{R(s)} = \frac{K_p.\theta_1}{s(1 + 2.Zeta.Tw.s + (Tw.s)^2) + K_p.\theta_2} . \tag{9}$$

In the same way, the closed-loop transfer function that relates the output Y_Y with the input R for the Y axis system is:

$$\frac{Y_Y(s)}{R(s)} = \frac{K_p.\theta_1}{s(1 + Tp1.s) + K_p.\theta_2} . \tag{10}$$

Equations 9 and 10 are the model-reference transfer functions used in the adaptive controllers MRAC. To complete the reference model for each axis, it is just necessary to add the delay from each plant model shown in equations 6 and 7, and given the appropriate initial values for the variable parameters θ_1 and θ_2 so that the reference models have the desired response for seismic signals.

3 Adaptive Controller Design

Once obtained the reference models, we can obtain the equations that describe the change of θ parameters by applying the adaptation MIT rule given in equation 2, but like the reference models transfer functions doesn't change with θ, the partial derivate of *Ym* respect to θ, this term can be omitted. Taken this into account, the adaptive algorithm can be obtained by:

$$\frac{\partial e}{\partial \theta}=\frac{\partial e}{\partial \theta_1}+\frac{\partial e}{\partial \theta_2}=\frac{\partial Y}{\partial \theta_1}+\frac{\partial Y}{\partial \theta_2}. \tag{11}$$

Equation 11 can be used for both axes, so the equations below are used to obtain the equations to describe the change of θ in the adaptive control.

$$\frac{\partial e}{\partial \theta_1}=\frac{\partial\left(\frac{K_p.\theta_1}{s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2}.R(s)\right)}{\partial \theta_1}\ . \tag{12}$$

$$\frac{\partial e}{\partial \theta_1}=\frac{K_p.R(s)}{s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2}\ . \tag{13}$$

$$\frac{\partial e}{\partial \theta_2}=\frac{\partial\left(\frac{K_p.\theta_1}{s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2}.R(s)\right)}{\partial \theta_2}\ . \tag{14}$$

$$\frac{\partial e}{\partial \theta_2}=\frac{-K_p.K_p.\theta_1.R(s)}{\left(s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2\right)^2}\ . \tag{15}$$

$$\frac{\partial e}{\partial \theta_2}=\frac{-K_p.Y(s)}{s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2}\ . \tag{16}$$

Based on equations 13 and 16, and by using the equation 2, the equation that describes the change of θ parameters to update the adaptive controller are:

$$\theta_1=-\frac{\gamma.e.R(s)}{s}\left(\frac{K_p}{s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2}\right)\ . \tag{17}$$

$$\theta_2=\frac{\gamma.e.Y(s)}{s}\left(\frac{K_p}{s(1+2.Zeta.Tw.s+(Tw.s)^2)+K_p.\theta_2}\right)\ . \tag{18}$$

It's important to notice that the reference models obtained in 9 and 10 don't include the plant delay, so the equation 17 also has to include this delay. In equation 18 it's not necessary because the term Y(s) already includes the delay since it corresponds to the real output plant. Also, in equations 17 and 18 and in the reference models equations 9 and 10 as well, all the parameters are fixed, and they correspond to the initial parameters values obtained by a tuning process of the system. These values are shown in table 2.

Table 2. Initial Adaptive Control Parameters

Parameters	X axis	Y axis
γ	0.0003	0.00035
θ_1	0.08	0.05
θ_2	0.08	0.05

Figure 5 shows the blocks diagram of the adaptive controller with Smith predictor used for both axes. Equations 19 to 22 are the final transfer functions for X axis and equations 23 to 26 the transfer functions for Y axis controllers.

$$Reference\_Model_X = \frac{6.619.\,e^{-0.0751s}}{0.0008484\,s^3 + 0.02337\,s^2 + s + 6.619}\ . \tag{19}$$

$$Plant\_without\_delay_X = \frac{82.74}{0.0008484\,s^3 + 0.02337\,s^2 + s}\ . \tag{20}$$

$$Model_X\_\theta_1 = \frac{82.74.\,e^{-0.0751s}}{0.0008484\,s^3 + 0.02337\,s^2 + s + 6.619}\ . \tag{21}$$

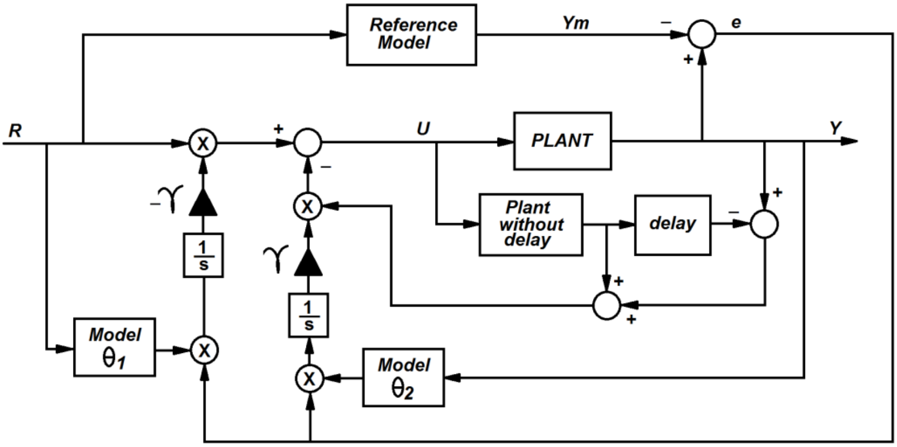


Fig. 5. Blocks Diagram of the Adaptive Control Implemented in X and Y axes

$$Model_X\_\theta_2 = \frac{82.74}{0.0008484\,s^3 + 0.02337\,s^2 + s + 6.619}\ . \tag{22}$$

$$Reference\_Model_Y = \frac{3.899.\,e^{-0.077s}}{0.02136\,s^2 + s + 3.899}\ . \tag{23}$$

$$Plant\_without\_delay_Y = \frac{77.98}{0.02136\,s^2 + s}\ . \tag{24}$$

$$Model_Y\_\theta_1 = \frac{77.98.\,e^{-0.077s}}{0.02136\,s^2 + s + 3.899}\ . \tag{25}$$

$$Model_Y\_\theta_2 = \frac{77.98}{0.02136\,s^2 + s + 3.899} \;. \tag{26}$$

4 Results

Once the controllers for X and Y axis were tuned using a simulation process, they were implemented on a 32 bits microcontroller platform by Microchip. This hardware is connected to a sever via serial connection through two XBee devices. The sample frequency used is 200 Hz, which is enough for the shaking table system used to generate the seismic signals, whose higher frequency components are up to 30 Hz. All the transfer functions were converted to difference equations in Z domain using a zero order hold and they were programmed in the hardware device.

Figure 6 shows the results for the test of the adaptive controllers implemented on the hardware device when the input signal is a periodic stair sequence. According to the figure, we can see the faster response adaptation of the system that follows the reference model in the second transient of the input signal.

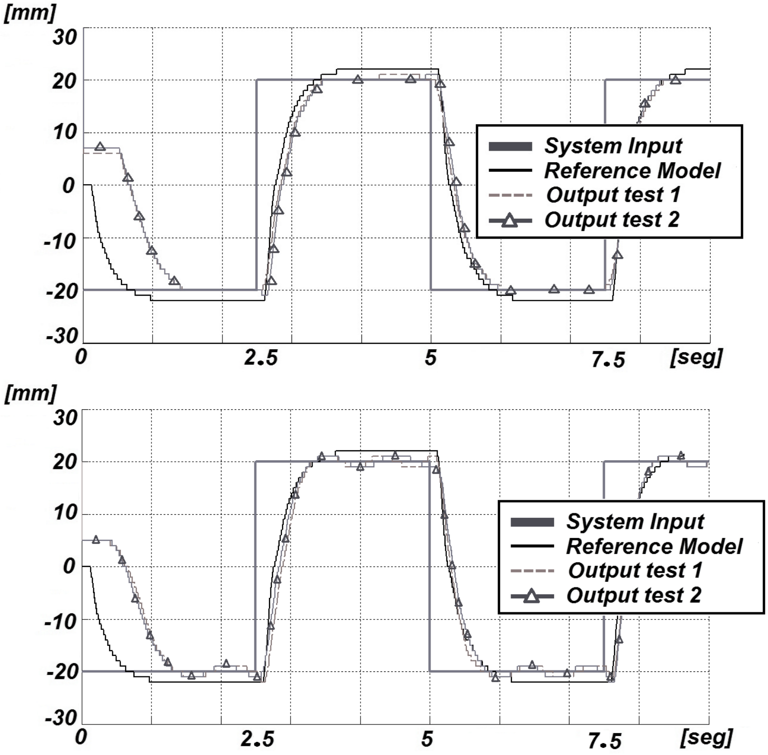


Fig. 6. Results for the Adaptive Controllers implemented on the hardware device. up) X-axis, down) Y-Axis.

The results presented in figure 6 shows that the adaptive controller response has high level approximation and repeatability for the different tests made on the shaking table. In this case, to analyze the approximation between the desired signal given by the reference model and the obtained signal we can use a measurement Normalized Root Mean Square Error – NRMSE, given in equation 27 [14], and the measurement of the amount of variability between the signals using the coefficient of determination R^2 described in equation 28 [15]. In both cases, the NRMSE and the R^2, two values are obtained, one for all the samples in the test and the other eliminating the first samples that takes the system to follow the reference model, that is approximately 2,5 seconds.

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i')^2}{\sum_{i=1}^{n}(x_i)^2}} \quad . \tag{27}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i')^2}{\sum_{i=1}^{n}(x_i - \mu_{xi})^2} = 1 - \frac{var(e)}{var(x_i)} \quad . \tag{28}$$

Table 3 shows the results obtained in 4 tests made on the shaking table with the periodic stair sequence for the X-axis, and Table 4 shows the same results for the Y-axis.

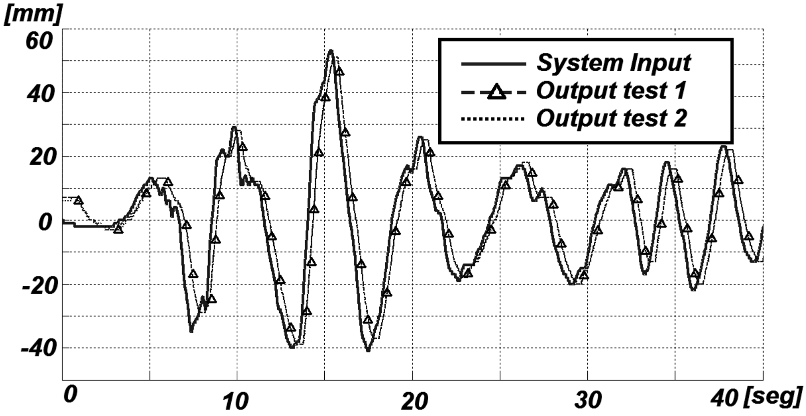
Table 3. NRMSE and R^2 results for X-axis

Test	All samples		Without first 2.5 seconds	
	NRMSE	R^2	NRMSE	R^2
1	0.20405	95.936	0.12073	98.536
2	0.20539	95.919	0.11419	98.689
3	0.21033	95.703	0.12475	98.435
4	0.21929	95.411	0.13463	98.191

Table 4. NRMSE and R^2 results for Y-axis

Test	All samples		Without first 2.5 seconds	
	NRMSE	R^2	NRMSE	R^2
1	0.19373	96.272	0.12689	98.412
2	0.18681	96.547	0.12339	98.484
3	0.19321	96.322	0.1298	98.313
4	0.2023	95.968	0.13867	98.073

Finally, figure 7 shows the results when the input of the system is a seismic signal, that for the test is the data collected from Imperial Valley, CA earthquake of October 15 1979. This earthquake was selected because has low frequency components, ideal for experiments in this shaking table, and was downloaded from the Center of Engineering Strong Motion Data (CESMD). However, in these results can be noticed that the Y-axis seismic tests have a slower adaptation time than for the X-axis. This is because the initial transitions of displacement in Y-axis have small values, lower than 10 millimeters unlike the X-axis.



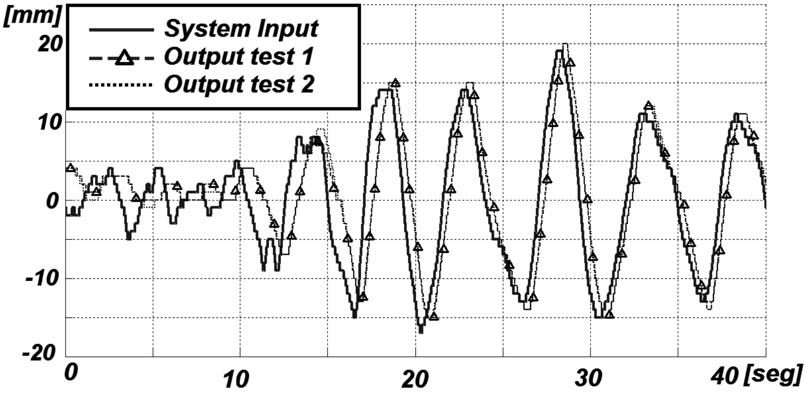


Fig. 7. Results for the Adaptive Controllers for seismic displacement signals. up) X-axis, down) Y-Axis

5 Conclusions

The results of this work show that the control system of a plant with high uncertainty and with delay can be done by implementing a reference model adaptive controller, for which it is only necessary to have a general knowledge of the dynamic behavior of the plant. In consequence, it was possible to implement an adaptive control algorithm with two adaptation parameters based on the rule MIT, modifying their structure to introduce a Smith predictor model which improves the performance of the controller by compensating the error generated in the difference equation due to the system delay.

Also, it was verified that the system delay can be handled outside of the closed-loop, so that the reference model can be easier to manipulate, obtaining an approximate response to those of the real system.

Moreover, although it wasn't mentioned in the document, the sensor resolution used for feedback is 1 millimeter, given by an encoder with 100 pulses per revolution. The response of the system is appropriate despite the error that is introduced by the rounding, obtaining ratios between the desired signals and the output signals above 90%. For this reason, the sensors will be improved for further works with the shaking table system.

On the other hand, the problem of the slow time for the adaptation in the Y-axis that appears for the amplitude of the input signal can be improved with the normalization of the MIT rule like is proposed by Astrom in [8], so that no matter what is the value of the input signal or the value for the adaptation gain value, the adaptive controller will always follow the reference model at the same speed.

Finally, the first test for the remote access to the system works properly over a LAN network, and currently it is validating over the RENATA network, through which it is intended to launch a digital control system laboratory based on the shaking table system.

References

1. Landau, I.D., Lozano, R., M'Saad, M., Karimi, A.: Adaptive Control: Algorithms, Analysis and Applications, 2nd edn. Springer (2011)
2. Tu, J.W., Jiang, S.J., Stoten, D.P.: The Seismic Response Reduction by Using Model Reference Adaptive Control Algorithm. In: 2010 International Conference on Mechanic Automation and Control Engineering (MACE), pp. 1215–1218. IEEE (2010)
3. Chen, J., Zhang, X., Tan, P., Wei, L.: Analysis of the Electro-hydraulic Servo Shaking Table with Flexible Payload. Technology, 708–713 (2009)
4. Seki, K., Iwasaki, M., Kawafuku, M., Hirai, H., Yasuda, K.: Adaptive Compensation for Reaction Force With Frequency Variation in Shaking Table Systems. IEEE Transactions on Industrial Electronics 56, 3864–3871 (2009)
5. Gavin, H.P., Hoagg, J.B.: Control Objectives for Seismic Simulators. Control, 3932–3937 (2009)
6. Seki, K., Iwasaki, M., Hirai, H.: Reaction Force Compensation With Frequency Identifier in Shaking Table Systems. Control, 673–678 (2010)
7. Rodríguez Rubio, F., López Sánchez, M.J.: Control Adaptativo y Robusto. Secretariado de Publicaciones de la Universidad de Sevilla (1996).
8. Astrom, K.J., Wittenmark, B.: Adaptive Control, 2nd edn. Addison-Wesley Longman Publishing Co. Inc. (1994)
9. Duraisamy, R., Dakshinamurthy, S.: An adaptive optimisation scheme for controlling air flow process with satisfactory transient performance. Maejo International Journal of Science and Technology 4, 221–234 (2010)
10. Swarnkar, P., Jain, S., Nema, R.K.: Effect of adaptation gain in model reference adaptive controlled second order system. Engineering, Technology &amp; Applied 1, 70–75 (2011)

11. Stellet, J.E.: Influence of Adaptation Gain and Reference Model Parameters on System Performance for Model Reference Adaptive Control. World Academy of Science, Engineering and Technology 60, 1768–1773 (2011)
12. Ioacnou, P.: Chapter 54. Model Reference Adaptive Control. In: The Control Handbook, pp. 847–858. CRC Press, IEEE Press (2000)
13. Normey-Rico, J.E., Camacho, E.F.: Predicción Para Control: Una Panorámica del Control de Procesos con Retardo. Revista Iberoamericana de Automática e Informática Industrial - RIAI 3, 5–25 (2006)
14. Frýza, T., Hanus, S.: Video signals transparency in consequence of 3D-DCT transform. In: Radioelektronika 2003 Conference Proceedings, pp. 127–130 (2003)
15. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers, 3rd edn. John Wiley & Sons, Ltd. (2003)

Implementation of a Single Chaotic Neuron Using an Embedded System

Luis González-Estrada, Gustavo González-Sanmiguel,
Luis Martin Torres-Treviño, and Angel Rodríguez

Universidad Autónoma de Nuevo León, FIME
Av. Universidad S/N
San Nicolás de los Garza, N.L.
{luis.gonzalezes,gustavo.gonzalezsn}@uanl.edu.mx,
{luis.torres.ciidit,angelrdz}@gmail.com

Abstract. A single chaotic neuron can be developed using a single neuron with a Gaussian activation function and feeding back the output to its input. The Gaussian activation function has two parameters, the center of mass and the width of the bell called sensibility factor. The change of these parameters determines the behavior of the single neuron that could be stationary but more of the times is dynamic inclusive chaotic. This paper presents an implementation of this single neuron in an embedded system generating a set of bifurcation plots illustrating the dynamic complexity that could have this simple system.

1 Introduction

Since the evolution of machines, it has been a desire of scientists to create a machine that can operate with increasing independence from human control in an unstructured and uncertain environment; such a machine could be called an autonomous, intelligent, or cognitive machine. Biological systems may be considered as a source of motivation and framework for the design of such autonomous machine. Biological neural control mechanisms are quite successful at dealing with uncertainty and complexity [1].

At engineering it is desirable generate computing structures that, similar to the brain, they can emulate complex behaviors and also can be controlled to create entities with superior abilities. A simple approximation to this objective can be made using a Single Recurrent Artificial Gaussian Neuron (SRAGN) which has a Gaussian activation function. The recurrence is given by the connection of the output to the input of the neuron. This simple connection provides the ability to exhibit stable, oscillating and chaotic responses according to its parameters [2,3,4].

Due to the social demands for sophisticated products and the requirements at the daily industrial processes to cover the human needs, is important the implementation on embedded systems of the new tools developed by scientific research. The embedded systems are often defined as a collection of programmable parts surrounded by ASICs (application-specific integrated circuits) and other

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 283–291, 2013.

standard components that interact continuously with an environment through sensors and actuators. The programmable parts include micro-controllers and DSP's (digital signal processors) [5,6].

This paper presents the implementation of a SRAGN in an embedded system based on the open-source platform Arduino, which is a micro-controller-based electronic board that uses the 8-bit microcontroller ATmega168 [7]. It is presented the algorithm used, the experimentation using two potentiometers to establish the parameters values and as results, some bifurcation diagrams are presented.

2 A Single Chaotic Neuron

A neuron is an information-processing unit that is fundamental to the operation of a neuronal network [8]. It is a model approximation of a biological neuron. The block diagram of Figure 1 shows the classical model of a neuron, here we identify three basic elements:

- A set of *synapses* or connecting links, each of which is characterized by a weight of its own. Specifically, an input signal X_i, connected to the neuron, is multiplied by the synaptic weight W_i.
- An *adder* for summing the input signals, pondered by the respective weight.
- An *activation function* for limiting the amplitude of the output of a neuron. Normally this function is no-linear such a Gaussian, Sigmoidal or Hyperbolic tangential forms.

In this paper we deal with a SRAGN in which, as can be seen in Figure 2, the output is connected to the main input, while the other two inputs control the parameters of the activation function.

The Gaussian activation function is given by (1) and has bell form as can be seen in Figure 3. Thus, the response, expressed in a discrete form, is governed by (2).

$$fa(x, \lambda, cm) = e^{-(\frac{x-cm}{\lambda})^2} \tag{1}$$

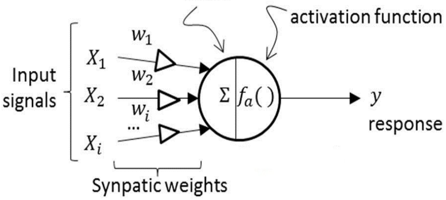


Fig. 1. Classical model of a neuron

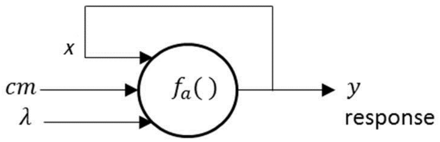


Fig. 2. Single Recurrent Artificial Gaussian Neuron

$$y_n = e^{-\left(\frac{y_{n-1}-cm}{\lambda}\right)^2} \tag{2}$$

Where:

- $n = 1, 2, 3, ...$ denotes each execution of the algorithm which pseudo-code in the next section. It is considered that the execution takes T seconds to complete the calculations.
- y_n is the neuron response in an instant.
- y_{n-1} is the neuron response T seconds before. With $y_{n-1} = 0$ when $n = 0$, i. e. y_0 is the initial condition.

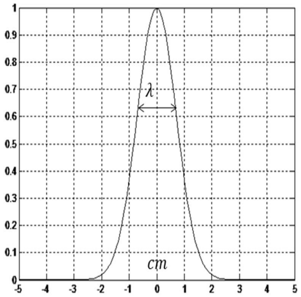
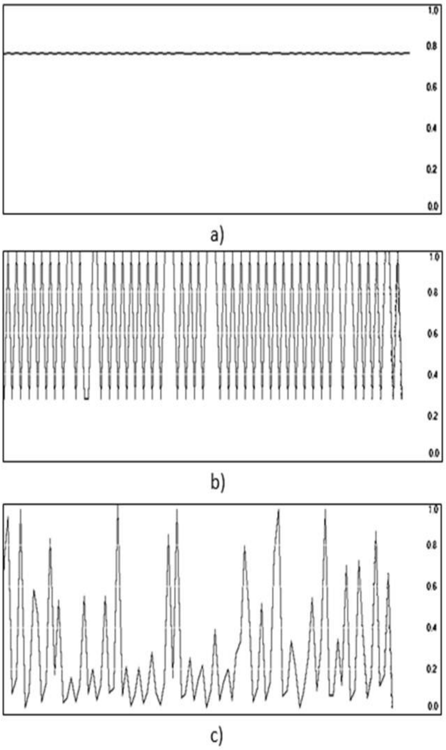


Fig. 3. Gaussian bell. Its two parameters are displayed in the plot.

As commented before, by changing the value of parameters λ and cm, the neuron can exhibit stable responses like in Figure 4.a, oscillating responses like in Figure 4.b and even chaotic responses like in Figure 4.c. Figure 4 shows an example of each of these behaviors in the output through time. The dynamics can be confirmed in the bifurcation diagrams at section 4.



a)

b)

c)

Fig. 4. a) Stable response, b) Oscillating response, c) Chaotic response

3 Implementation of Single Neurons in an Embedded System

Arduino is an open-source platform to work with electronic, control, robotic and informatics prototypes. The platform consists in a microcontroller-based board and an Integrated Development Environment (IDE) software to write, debug and download programs to the board. Both the board and the IDE are opened to the users, either hardware or software modifications can be made. There are some versions of this board according to the desired features; the implementation presented in this paper uses the Arduino Diecimila one, which is based on the 8-bit microcontroller ATmega168 which main features are: operation at 5Vdc, 16 MHz clock speed, 16 KB FLASH memory program and 1 KB SRAM data memory.

This board is the main component of the prototype and has in its memory the complete neuron working. Also were connected two potentiometers to modify the value of the parameters, each one in the range of [0, 5] Vdc that are converted to a [0, 1] floating point normalized range by doing this:

1. Lecture in a [0, 5] Vdc range.
2. Analog to Digital Converter (ADC) gets [0, 255] binary.
3. Divide by 255.0 gets [0, 1] floating point normalized range.

So, the calculation of the neuron response is in the [0, 1] normalized range but this value is converted to a [0, 5] Vdc representation that can be measured by an instrument such as an oscilloscope by doing this:

1. Variable in a [0, 1] floating point normalized range.
2. Multiply by 255.0 gets [0, 255] binary.
3. Pulse Width Modulation (PWM) delivers an average voltage signal in the range [0, 5] Vdc.

Figure 5 shows a schematic diagram of the system implementation. ADC and PWM modules are implemented internally in the microcontroller. The pins used for these modules are mapped on board terminals.

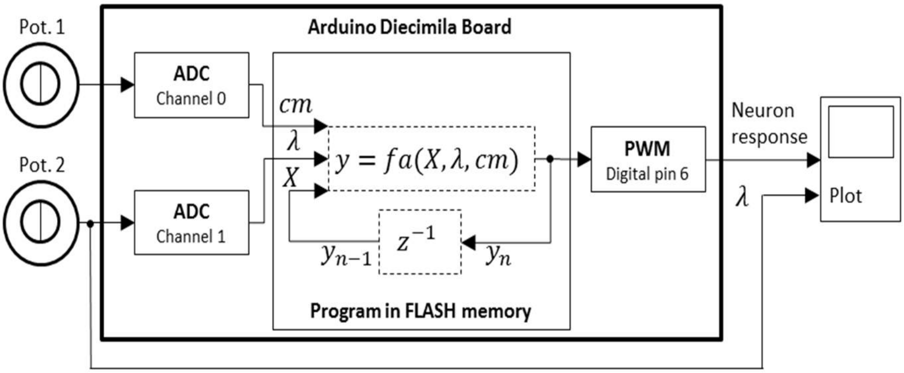


Fig. 5. Shematic diagram of the system

The program was written using the programming language provided by the Arduino Platform and the compiler was responsible for the code optimization and translation to machine code. The ATmega168 microcontroller includes native instructions to work with signed/unsigned/fraction/no-fraction multiplications and 8-bit number comparisons which make this microcontroller a good option to work with. Now, here is presented the pseudo-code of the program, it is easy to see that this program can be extended to more than one SRAGN.

Pseudo-code:

```
Assignation of the initial condition $y_{n-1} = 0$
Do an infinite loop with containing these 5 steps:
1. Read data: read $x$, $\lambda$ and $cm$ from the analog input pins of the board.
2. Compute neuron response using equation (2). $y_n = fa(x, \lambda, cm)$. Notice that if $\lambda$=0
then the division in (2) renders undefined, so assignment must be $y_n = 1$
3. Write the value of the response in an analog output pin of the board.
4. Update the value of the previous response. $y_{n-1} = y_n$
5. Ensure delay enough time to complete T seconds since the loop execution began.
```

It is important to notice that the feedback of the output is made by software instead to connect a wire from output pin to an input pin. This action avoids noise and improves the calculation. Finally, note that around 300 or even more neurons can be processed in this embedded system.

4 Experimental Results

In this section there are presented some bifurcation diagrams. A personal computer was used to acquire the data from the embedded system and make the plots. All bifurcation diagrams are made using the output versus the change of the parameter λ while cm remains fixed in each diagram. The value of cm

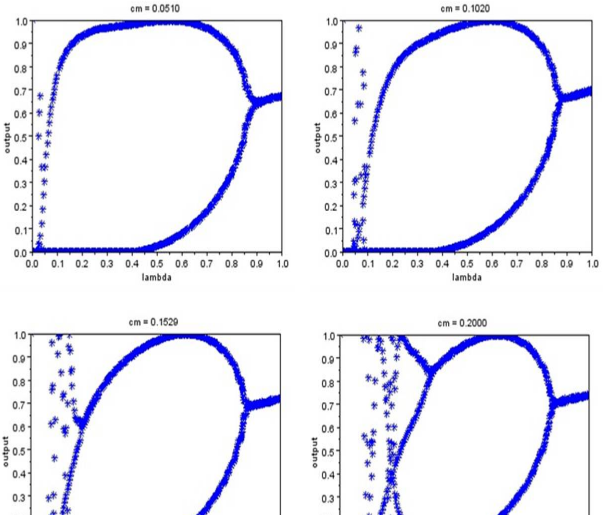


Fig. 6. Bifurcation diagram using cm = 0.0510, 0.1020, 0.1529 and 0.2000

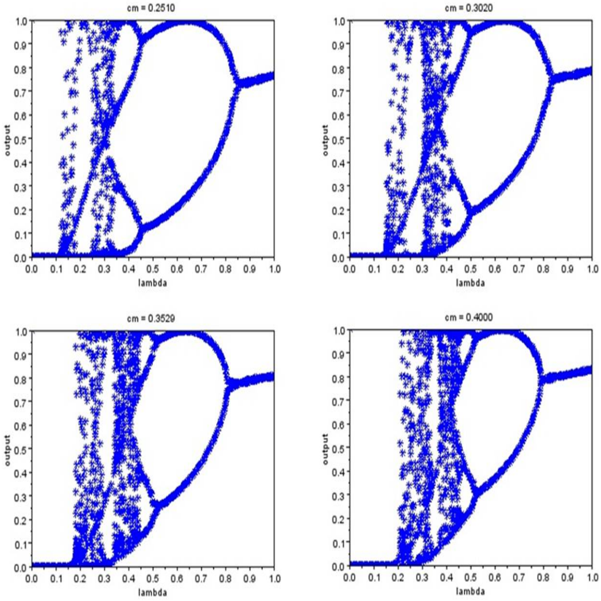


Fig. 7. Bifurcation diagram using cm = 0.2510, 0.3020, 0.3529 and 0.4000

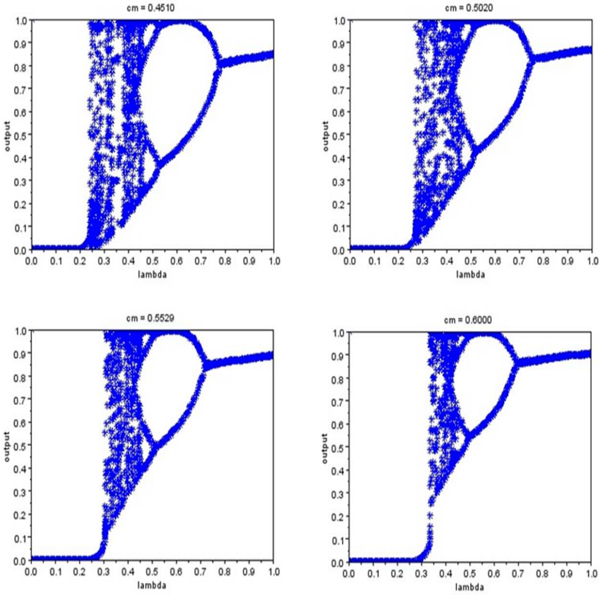


Fig. 8. Bifurcation diagram using cm = 0.4510, 0.5020, 0.5529 and 0.6000

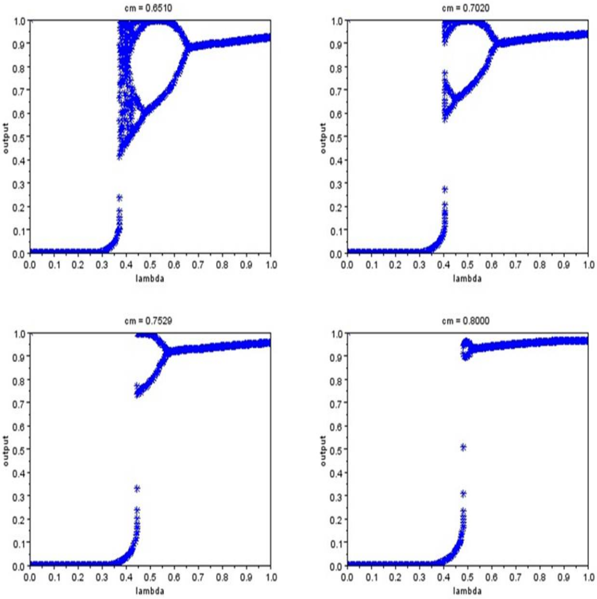


Fig. 9. Bifurcation diagram using cm = 0.6510, 0.7020, 0.7529 and 0.8000

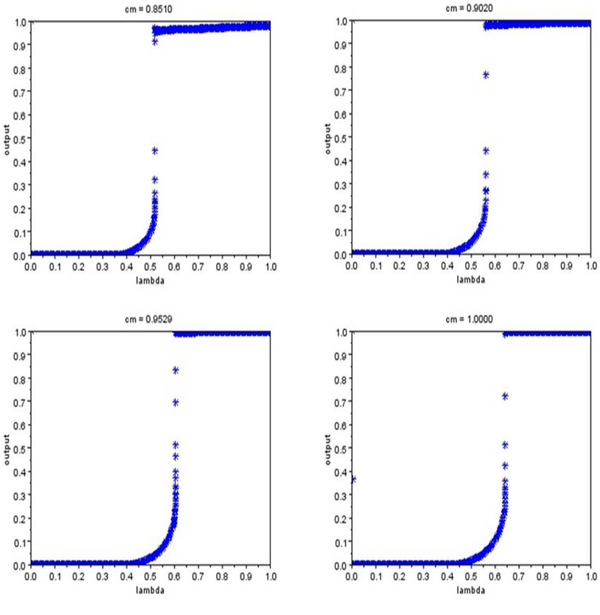


Fig. 10. Bifurcation diagram using cm = 0.8510, 0.9020, 0.9529 and 1.0000

starts at 0.05 until 1.0 with increments of approximately 0.05 because the 8-bit precision defined by the PWM and ADC modules. The period of execution T has been fixed to 10 milliseconds.

5 Conclusions

An interpretation for these bifurcation diagrams could be that, for each value of cm, λ chooses the type of response: stable, oscillating or chaotic. For small values in cm, the predominant response (in all λ range) tends to be oscillating, for medium values in cm the predominant response is chaotic and finally for bigger values in cm (close to 1), the predominant response is stable. By "predominant response" we mean to the graphic form in the greater part of the range of λ, for example, in the last plot (cm=1) we can see that the predominant response is stable, however the general form is something like a unitary pulse, i. e. a bi-stable non-linear system.

A multiple interaction of these neurons could make interesting structures that have complex behaviors, so the future work is to study more this neuron, what is the cause of these behaviours, the structures that can be formed and also the control or adjustment of the parameters to get desirable behaviors, for example, the representation of oscillations observed in heart pulses, the activation of muscles for walking or breathing, etc.

Acknowledgments. We thank CONACYT by the support provided in this project (grant number 258615 and grant number 258684) also thanks to the support provided by PROMEP/103.5/10/6687 and PAICYT-UANL grant number IT562-10.

References

1. Rao, D.H., Gupta, M.M.: Neuro-Control Systems, Theory and Applications. IEEE Press (1994)
2. no, L.T.T.: Chaos in a single recurrent artificial neuron. Research in Computing Science 28, 49–56 (2007)
3. Jung, J.K.J.: Fully integrated circuit design aiharas chaotic neuron model, pp. 258–259. IEEE Conference Publications (2010)
4. Hsu, C.C., Gobovic, D.: Chaotic neuron models and their vlsi circuit implementations. IEEE Transactions on Neural Networks 7, 1339–1350 (1996)
5. Vahid, F., Gajski, D.D.: Specification and design of embedded software-hardware systems. IEEE Design and Test of Computers 12, 53–67 (1995)
6. Giusto, F.B.M.C.P.: Hardware-Software co-design of embedded systems. Kluwer Academic Publishers (1997)
7. Massimo Banzi, D.C.: (Arduino official web page), `http://www.arduino.cc`
8. Haykin, S.: Neural Networks, A comprehensive Foundation. Prentice-Hall (1999)

Application of a Method Based on Computational Intelligence for the Optimization of Resources Determined from Multivariate Phenomena

Angel Kuri-Morales

Departamento de Computación
Instituto Tecnológico Autónomo de México
Mexico City, Mexico
akuri@itam.mx

Abstract. The optimization of complex systems one of whose variables is time has been attempted in the past but its inherent mathematical complexity makes it hard to tackle with standard methods. In this paper we solve this problem by appealing to two tools of computational intelligence: a) Genetic algorithms (GA) and b) Artificial Neural Networks (NN). We assume that there is a set of data whose intrinsic information is enough to reflect the behavior of the system. We solved the problem by, first, designing a system capable of predicting selected variables from a multivariate environment. For each one of the variables we trained a NN such that the variable at time *t+k* is expressed as a non-linear combination of a subset of the variables at time *t*. Having found the forecasted variables we proceeded to optimize their combination such that its cost function is minimized. In our case, the function to minimize expresses the cost of operation of an economic system related to the physical distribution of coins and bills. The cost of transporting, insuring, storing, distributing, etc. such currency is large enough to guarantee the time invested in this study. We discuss the methods, the algorithms used and the results obtained in experiments as of today.

Keywords: Computational intelligence, forecasting, optimization, non-linear modeling.

1 Introduction

The problem of optimizing the future assignment of resources to a dynamic system whose behavior displays a non-chaotic behavior poses two clearly distinct challenges: a) One has to be able to reliably predict the future behavior of the system based on its past history and b) Assuming the purported prediction's reliability, one has to establish the best possible values of the variables involved such that a cost function is minimized. In practical terms this means that, from a table of values of a set of variables $V(t)=v_1(t),...,v_n(t)$ reflecting the previous known values of a system $\boldsymbol{S(t)=f(V(t))}$ at time t with an associated function of cost $C(t) = f(V(t))$ we are to minimize $C(t+k) = f(V(t+k))$ where k represents the time displacement relative to the present time t which we want forecast.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 292–303, 2013.

To do this, then, we first want to find $V(\tau) = \{v_1(\tau),...,v_n(\tau)\}$ such that

$$v_i(\tau) = f_i(v_1(t),...,v_n(t)) \quad \text{for } i = 1, \ldots, n \tag{1}$$

(where, for convenience, we have made $\tau = t + k$). Second, we would like to minimize $C(\tau)$.

To exemplify, assume that $C(\tau)$ has the polynomial form

$$C(\tau) = c_{0000} + c_{0123} v_2(\tau) v_3(\tau)^2 v_4(\tau)^3 + c_{3211} v_1(\tau)^3 v_2(\tau)^2 v_3(\tau) v_4(\tau) \tag{2}$$

The form of (1) is determined from the conditions of the problem. Generally, it is a combination of the conditions of the dynamics of the system and normally does not have a closed mathematical formulation. This is one of the main reasons behind the choice of a GA as a minimization tool.

Our first aim is to find the functions which yield the approximate values of $v_i(\tau)$ for $i=1, 2, 3, 4$. Such forecasted values (call them $v_i{}^*(\tau)$) will have an expected approximation error ε_i such that $v_i{}^*(\tau) = v_i(\tau) \pm \varepsilon_i$. Our second aim is to minimize $C(\tau)$ where, because of the expected approximation errors, the following constraints do hold: $v_i{}^*(\tau) - \varepsilon_i \le v_i(\tau) \le v_i{}^*(\tau) + \varepsilon_i$. This minimization process will deliver the values which will yield the forecasted minimum value of $C(\tau)$ taking into consideration possible uncertainties in the forecasted values of the $v_i(t)$.

We have approximated the $v_i(\tau)$ using a set of multi-layer perceptron NNs [1] and minimized $C(\tau)$ using a GA [2]. The unknown values of the model of $v_i(t)$ loosely correspond to the weights in the NN; one NN for each of the $v_i(\tau)$. We denote the function implemented by a NN for $v_i(\tau)$ by

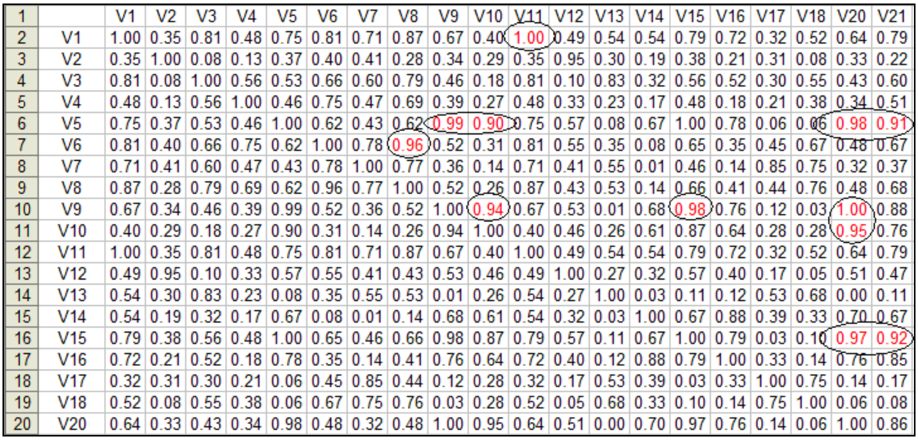
$$v_i{}^*(\tau) = NN_i(v_1(t),...,v_n(t)) \tag{3}$$

To achieve our goals we must solve a number of issues such as: a) Which variables to consider in our problem, b) How to pre-condition the values of the v_i's, c) How to set the data to achieve the desired forecast, d) What is the architecture of the NN_i (That is, i) The number of layers in the NNs, ii) The number of nodes (*neurons*) in the hidden layer, iii) The activation function in each of the layers, etc.), e) How to measure the NN_i's reliability, f) What GA to apply (and its settings) during the optimization process.

In part 2 we discuss the pre-conditioning of the data. In part 3 we describe the way the NNi's were set, in part 4 we describe the training method for the NN_i, in part 5 we describe the optimization process, in part 6 we describe some experimental results, in part 7 we offer our conclusions.

2 Data Set Up

In what follows we describe the nature of the data with which we worked, the way we set it up to achieve the desired forecasting and the pre-conditioning which was required to increase our probabilities of success. The way in which historical data may be applied to generate reliable forecasting may be better explained by taking an example. To begin with, a correlation matrix was used to detect those variables which exhibited a correlation larger than 90% as shown in figure 1.



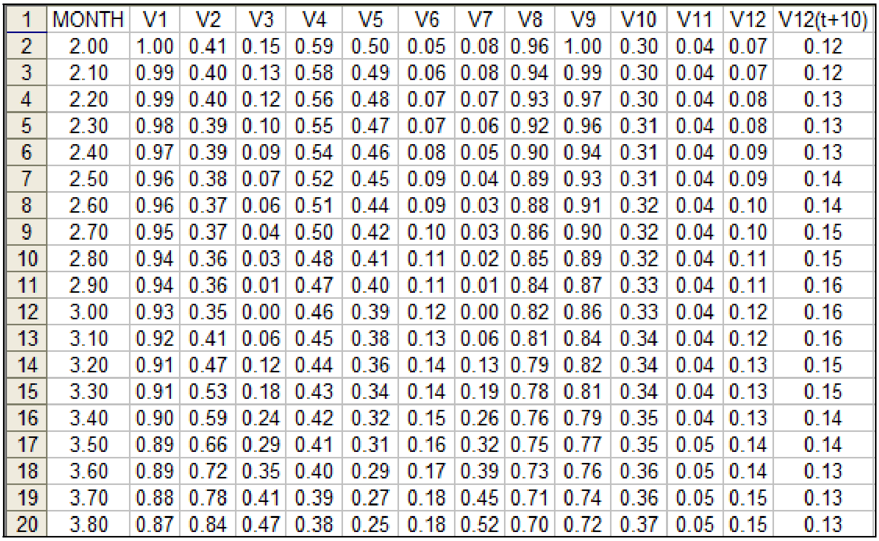
1		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V20	V21
2	V1	1.00	0.35	0.81	0.48	0.75	0.81	0.71	0.87	0.67	0.40	1.00	0.49	0.54	0.54	0.79	0.72	0.32	0.52	0.64	0.79
3	V2	0.35	1.00	0.08	0.13	0.37	0.40	0.41	0.28	0.34	0.29	0.35	0.95	0.30	0.19	0.38	0.21	0.31	0.08	0.33	0.22
4	V3	0.81	0.08	1.00	0.56	0.53	0.66	0.60	0.79	0.46	0.18	0.81	0.10	0.83	0.32	0.56	0.52	0.30	0.55	0.43	0.60
5	V4	0.48	0.13	0.56	1.00	0.46	0.75	0.47	0.69	0.39	0.27	0.48	0.33	0.23	0.17	0.48	0.18	0.21	0.38	0.34	0.51
6	V5	0.75	0.37	0.53	0.46	1.00	0.62	0.43	0.62	0.99	0.90	0.75	0.57	0.08	0.67	1.00	0.78	0.06	0.06	0.98	0.91
7	V6	0.81	0.40	0.66	0.75	0.62	1.00	0.78	0.96	0.52	0.31	0.81	0.55	0.35	0.08	0.65	0.35	0.45	0.67	0.48	0.67
8	V7	0.71	0.41	0.60	0.47	0.43	0.78	1.00	0.77	0.36	0.14	0.71	0.41	0.55	0.01	0.46	0.14	0.85	0.75	0.32	0.37
9	V8	0.87	0.28	0.79	0.69	0.62	0.96	0.77	1.00	0.52	0.26	0.87	0.43	0.53	0.14	0.66	0.41	0.44	0.76	0.48	0.68
10	V9	0.67	0.34	0.46	0.39	0.99	0.52	0.36	0.52	1.00	0.94	0.67	0.53	0.01	0.68	0.98	0.76	0.12	0.03	1.00	0.88
11	V10	0.40	0.29	0.18	0.27	0.90	0.31	0.14	0.26	0.94	1.00	0.40	0.46	0.26	0.61	0.87	0.64	0.28	0.28	0.95	0.76
12	V11	1.00	0.35	0.81	0.48	0.75	0.81	0.71	0.87	0.67	0.40	1.00	0.49	0.54	0.54	0.79	0.72	0.32	0.52	0.64	0.79
13	V12	0.49	0.95	0.10	0.33	0.57	0.55	0.41	0.43	0.53	0.46	0.49	1.00	0.27	0.32	0.57	0.40	0.17	0.05	0.51	0.47
14	V13	0.54	0.30	0.83	0.23	0.08	0.35	0.55	0.53	0.01	0.26	0.54	0.27	1.00	0.03	0.11	0.12	0.53	0.68	0.00	0.11
15	V14	0.54	0.19	0.32	0.17	0.67	0.08	0.01	0.14	0.68	0.61	0.54	0.32	0.03	1.00	0.67	0.88	0.39	0.33	0.70	0.67
16	V15	0.79	0.38	0.56	0.48	1.00	0.65	0.46	0.66	0.98	0.87	0.79	0.57	0.11	0.67	1.00	0.79	0.03	0.10	0.97	0.92
17	V16	0.72	0.21	0.52	0.18	0.78	0.35	0.14	0.41	0.76	0.64	0.72	0.40	0.12	0.88	0.79	1.00	0.33	0.14	0.76	0.85
18	V17	0.32	0.31	0.30	0.21	0.06	0.45	0.85	0.44	0.12	0.28	0.32	0.17	0.53	0.39	0.03	0.33	1.00	0.75	0.14	0.17
19	V18	0.52	0.08	0.55	0.38	0.06	0.67	0.75	0.76	0.03	0.28	0.52	0.05	0.68	0.33	0.10	0.14	0.75	1.00	0.06	0.08
20	V20	0.64	0.33	0.43	0.34	0.98	0.48	0.32	0.48	1.00	0.95	0.64	0.51	0.00	0.70	0.97	0.76	0.14	0.06	1.00	0.86

Fig. 1. Correlation matrix

When this happens only one (it does not matter which and this election is left to the user) is retained reducing the number of independent variables (in this example, from 20 to the 12 shown in figure 2). For simplicity, the values have been re-labeled. Assume, then, that we have such table of values. There we have ordered the variables (V_1, … , V_{12}) according to the date of registration. The oldest data are shown first; the newest last. What we would like to do is forecast (in this example) the values of V_{12} for $k=10$. That is, we want to find a function of the form

$$V_{12}(t+10) = f(V_1(t),...,V_{12}(t)) \tag{4}$$

To do this we simply set the values of a (new) dependent variable V_{12} at time $t+10$ as a function of the variables V_i at time t as shown in figure 2. The values of column V_{12} are repeated on $V_{12}(t+10)$ starting at the 12-th row. Notice that all variables have been scaled to the interval [0,1]. This is done to avoid the induction of a variable's spurious importance because of different measurement units. It also has to do with the activation function of the perceptrons we used. We applied a *tanh* function. It is symmetric to the *X* axis and has maxima at ± 1, as shown in figure 3.



1	MONTH	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V12(t+10)
2	2.00	1.00	0.41	0.15	0.59	0.50	0.05	0.08	0.96	1.00	0.30	0.04	0.07	0.12
3	2.10	0.99	0.40	0.13	0.58	0.49	0.06	0.08	0.94	0.99	0.30	0.04	0.07	0.12
4	2.20	0.99	0.40	0.12	0.56	0.48	0.07	0.07	0.93	0.97	0.30	0.04	0.08	0.13
5	2.30	0.98	0.39	0.10	0.55	0.47	0.07	0.06	0.92	0.96	0.31	0.04	0.08	0.13
6	2.40	0.97	0.39	0.09	0.54	0.46	0.08	0.05	0.90	0.94	0.31	0.04	0.09	0.13
7	2.50	0.96	0.38	0.07	0.52	0.45	0.09	0.04	0.89	0.93	0.31	0.04	0.09	0.14
8	2.60	0.96	0.37	0.06	0.51	0.44	0.09	0.03	0.88	0.91	0.32	0.04	0.10	0.14
9	2.70	0.95	0.37	0.04	0.50	0.42	0.10	0.03	0.86	0.90	0.32	0.04	0.10	0.15
10	2.80	0.94	0.36	0.03	0.48	0.41	0.11	0.02	0.85	0.89	0.32	0.04	0.11	0.15
11	2.90	0.94	0.36	0.01	0.47	0.40	0.11	0.01	0.84	0.87	0.33	0.04	0.11	0.16
12	3.00	0.93	0.35	0.00	0.46	0.39	0.12	0.00	0.82	0.86	0.33	0.04	0.12	0.16
13	3.10	0.92	0.41	0.06	0.45	0.38	0.13	0.06	0.81	0.84	0.34	0.04	0.12	0.16
14	3.20	0.91	0.47	0.12	0.44	0.36	0.14	0.13	0.79	0.82	0.34	0.04	0.13	0.15
15	3.30	0.91	0.53	0.18	0.43	0.34	0.14	0.19	0.78	0.81	0.34	0.04	0.13	0.15
16	3.40	0.90	0.59	0.24	0.42	0.32	0.15	0.26	0.76	0.79	0.35	0.04	0.13	0.14
17	3.50	0.89	0.66	0.29	0.41	0.31	0.16	0.32	0.75	0.77	0.35	0.05	0.14	0.14
18	3.60	0.89	0.72	0.35	0.40	0.29	0.17	0.39	0.73	0.76	0.36	0.05	0.14	0.13
19	3.70	0.88	0.78	0.41	0.39	0.27	0.18	0.45	0.71	0.74	0.36	0.05	0.15	0.13
20	3.80	0.87	0.84	0.47	0.38	0.25	0.18	0.52	0.70	0.72	0.37	0.05	0.15	0.13

Fig. 2. Original scaled data

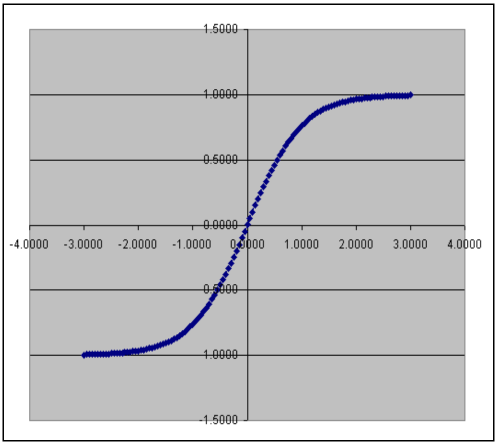


Fig. 3. Hyperbolic Tangent

3 Neural Network Architecture

We now set the NN which will approximate (4) as shown in figure 4. The number of input neurons is determined by the problem and corresponds to the number of input variables. The number of output neurons corresponds to the number of variables to be predicted. Rather than finding a single network for all the desired outputs (in this case 12) we decided to find one NN for each variable thus needing 12 networks. The number of layers (in this case 3) is determined by considering that any three-layered network is sufficient to approximate a continuous function [3]. We guaranteed such continuous behavior by interpolating with a natural spline [4]. Every neuron in the network consists of an adder whose output is fed to an activation function as shown in figure 5. In our models all activation functions in the input and output layers are linear while the ones in the hidden layer are *tanh*. This configuration has been shown [14] to work properly for regression problems such as this. The number of neurons in the hidden layer was determined from the heuristic

$$H \approx (S - 3O)/[3(I + O + 1)] \tag{5}$$

where H = number of neurons in the hidden layer, S = number of objects in the data set, I/O = number of neurons in the input/output layer. The intuition behind (5) is that the number of connections should be enough for the weights to express the information present in the data but not so many as to deprive the NN of its generalization properties. Therefore, they should be approximately equal to one third the number of elements in the data. From (5) we determined that *H=3*.

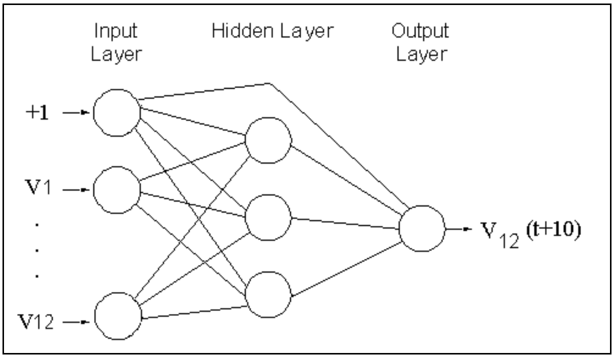


Fig. 4. A Neural Network for $V_{12}(t+10)$

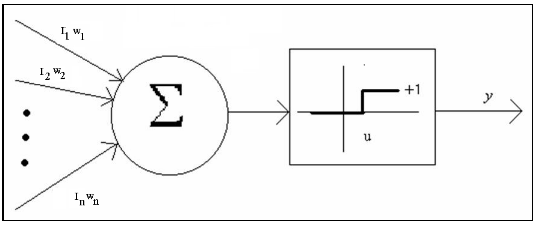


Fig. 5. A Perceptron

4 Training the Neural Networks

To train out NN's we used a GA rather than the classical backpropagation algorithm [5], [6]. The GA we used was not the Simple GA postulated by Holland but the so-called Eclectic GA (EGA) proposed in [7]. Use of the EGA frees the designer of the NN from the specification of the parameters associated to the BPA such as learning rates and moments. The values of these four parameters (2 for each of the H and O layers) are difficult to determine and have a definite impact on the behavior of the training phase [8]. We represent every weight as a fixed point signed binary number. The structure of the chromosome used to train the NN is shown in figures 6 and 7.

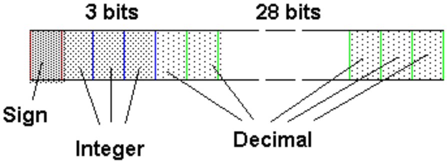


Fig. 6. Representation of one weight

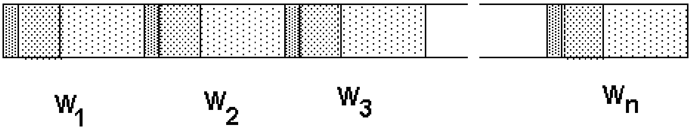
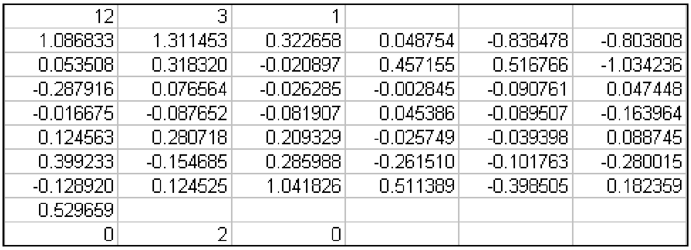


Fig. 7. Full Genome for the Neural Network

Furthermore, by training the NN with EGA we are able to select any measure of error to determine the suitability of the calculated weight. We implemented the following four norms: $L_e = \sum 10^{|\varepsilon_i|}$, $L_1 = \sum |\, \varepsilon_i \,|$, $L_2 = \sum \varepsilon_i^2$ and $L_\infty = \max(|\, \varepsilon_i \,|)$; where by ε_i we denote the approximation error for the *i-th* object of the sample. Each one of these norms may be used to determine the values of the weights by minimizing the ε_i 's relative to the whole data set. On the other hand, BPA, as is well known, is restricted to L_2. The data set was divided into a training set (TRS) and a test set (TSS). TRS consisted of a uniform random sample of 75% of the objects in the data set. TSS consisted of the remaining objects. During the training phase, the NN's weights were evolved with EGA for 500 epochs. For every 25 epochs the NN was tested vs. TSS. The weights corresponding to best error in the test phase were kept. Therefore, the model is the one whose performance is the best for TSS and not for TRS. This strategy ensures that the NN will have the best possible predictive capabilities *outside* the known data. This strategy corresponds to what is usually called cross-validation. Hence, the resulting model is completely described by a) The number of neurons in layers I-H-O, b) The weights of the connections between the layers and c) The kind of activation functions. In figure 8 the model for $V_{12}(t+10)$ is shown. The first line shows the number of neurons in layers $\mathrm{I-H-O}$. The last line describes the activation functions for layers $\mathrm{I-H-O}$; where 0 means "linear", 1 means "logistic" and 2 stands for "tanh". In this NN there are (I+1)H (39) connections from layer I to layer H; H+1 (4) connections from layer H to layer O. Thus, the first 39 weights shown correspond to the I-H connections; the last four to the H-O connections.



12	3	1			
1.086833	1.311453	0.322658	0.048754	-0.838478	-0.803808
0.053508	0.318320	-0.020897	0.457155	0.516766	-1.034236
-0.287916	0.076564	-0.026285	-0.002845	-0.090761	0.047448
-0.016675	-0.087652	-0.081907	0.045386	-0.089507	-0.163964
0.124563	0.280718	0.209329	-0.025749	-0.039398	0.088745
0.399233	-0.154685	0.285988	-0.261510	-0.101763	-0.280015
-0.128920	0.124525	1.041826	0.511389	-0.398505	0.182359
0.529659					
0	2	0			

Fig. 8. The model for $V_{12}(t+10)$

The models corresponding to variables *$V_i(t+10)$* individually take the place of the $v_i(\tau) = f_i(v_1(t),\ldots,v_n(t))$ of equation (1). Notice that, because of cross validation, we are able to determine the approximation error of our models. In figure 9 we compare the actual values of V_{12} and those of NN_{12}. In figure 10 we show the approximation errors in NN_{12}. Notice the scale. The average of the percentage error ε_i was 0.011958 while the largest ε_i was 0.073768. Hence, as anticipated, we have an uncertainty of $\approx \pm 0.074$ on the predicted values. The simplified cost function $C(\tau)$ for this system was determined, from the problem's conditions, to be

$$C(\tau) = 2.35v1(\tau)v3(\tau) + 6.17v5(\tau)v12(\tau) + 3.4v12(\tau)v5(\tau) - \quad . \\ 2v1(\tau)v3(\tau)v12(\tau) + 154 \qquad (6)$$

Notice that not all of the *$v_i(t)$* participate in equation (6) directly. However, they do form part of the functions for *$NN_j(t)$* for *$j=1,3,5,12$* since *$NN_i(t)=f(v_1(t),\ldots,v_{12}(t))$*. Accordingly, we determined the models for *NN_1*, *NN_3*, *NN_5* and *NN_{12}*.

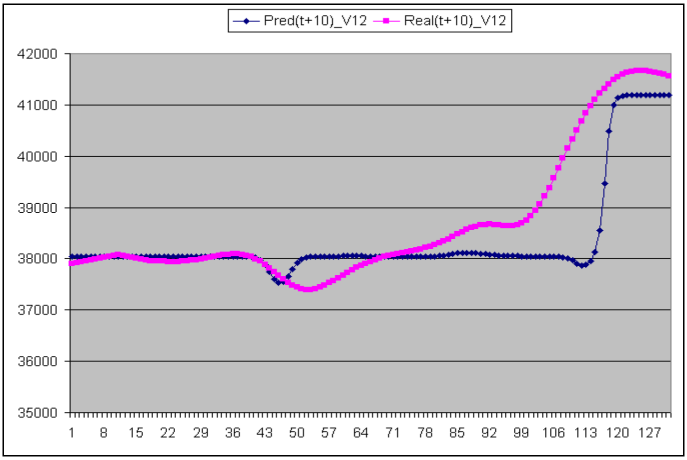


Fig. 9. Real vs. predicted values for $V_{12}(t+10)$

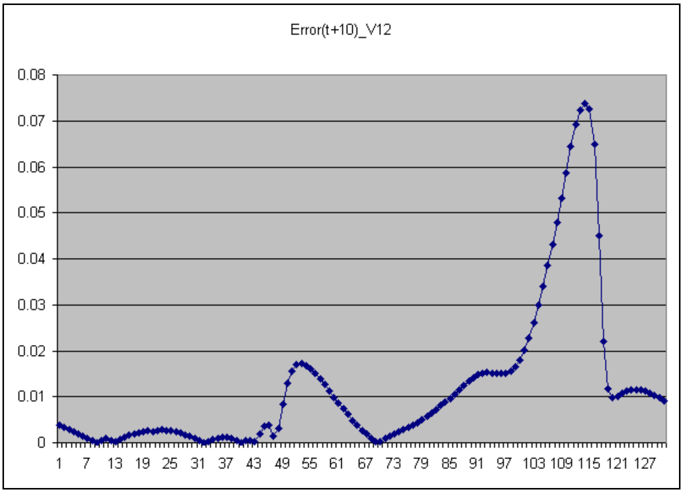


Fig. 10. Error values for V_{12}(t+10)

The real vs. predicted values for $V_1(\tau)$, $V_3(\tau)$ and $V_5(\tau)$ are shown in figures 11, 12 and 13 respectively. From the NN models we calculated $\varepsilon 1 = \pm 0.031$, $\varepsilon 3 = \pm 0.072$, $\varepsilon 5 = \pm 0.012$ and, as already pointed out, $\varepsilon 12 \approx \pm 0.074$.

5 Optimizing the Cost Function

Figures 7, 9, 10 and 11 give a graphical feeling of the good fit of the forecasting NNs. And the worst case error, although not necessarily homogeneous for all the values, is always less than 8%. In any case, this possible error, small as it may be, introduces an inherent uncertainty on the values of the target variables V_1, V_3, V_5 and V_{12}. To take into consideration the possible errors we proceeded to minimize equation (6) with the constraints included below and for those values predicted by NN_i.

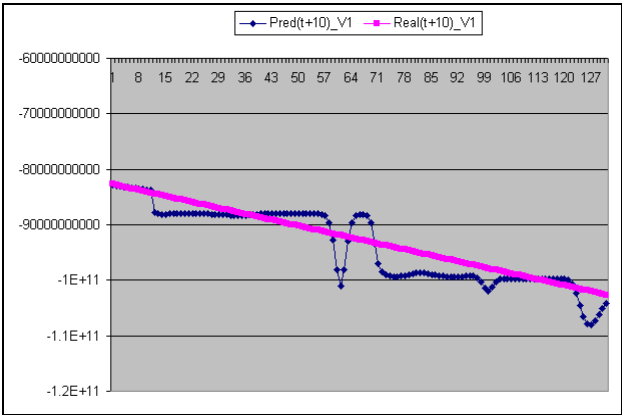


Fig. 11. Real vs. predicted values for V_1(t+10)

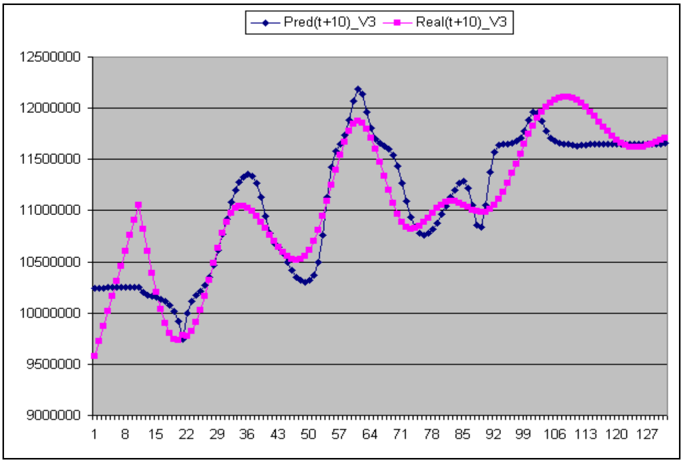


Fig. 12. Real vs. predicted values for V3(t+10)

In table 1 we have included the values of the $V_i(t+1)$ for $i=1,3,5,12,$ as per $NN_i(t)$. We also include the ε_i from which the lower and upper values of the predicted variables will be inferred:

$$V_i*(t+10)-\varepsilon_i \le V_i(t+10) \le V_i*(t+10)+\varepsilon_i$$

Table 1. Forecasted values en percentage errors for $t+10$

i	$V_i*(t+10)$	ε_i
1	*-1,026.24*	*0.031*
3	*116.975*	*0.072*
5	*1,882.04*	*0.012*
12	*415.70*	*0.074*

From table 1 we may calculate the constraints that apply to the minimization process of equation (6). The full minimization problem, with its restrictions, is shown next.

Minimize

$$C(\tau)=2.35v1(\tau)v3(\tau)+6.17v5(\tau)v12(\tau)+3.4v12(\tau)v5(\tau)- \\ 2v1(\tau)v3(\tau)v12(\tau)+154 \quad .$$

Subject to

1) $-1,058.04 \le V_1(\tau) \le -994.42$
2) $108.60 \le V_2(\tau) \le 125.40$
3) $1861.4 \le V_5(\tau) \le 1906.65$
4) $384.93 \le V_{12}(\tau) \le 446.45$

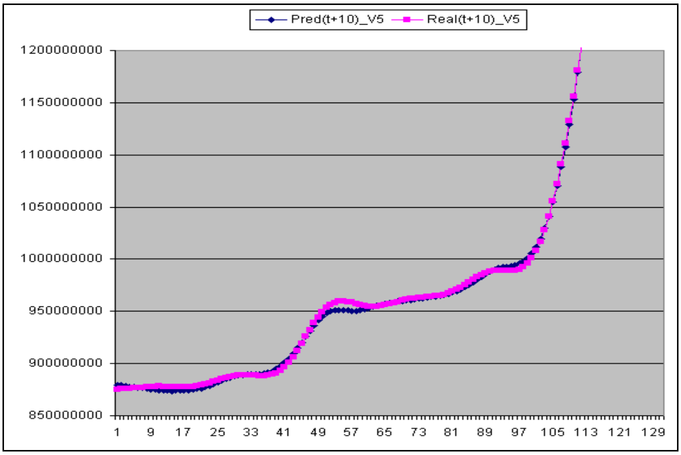


Fig. 13. Real vs. predicted values for V5(t+10)

In order for a GA to tackle constrained problems, it is necessary to define a penalty function [9]. We used the method described in [10]. It consists of defining the penalty function as follows:

$$P(\vec{x}) = \begin{cases} \left[K - \sum_{i=1}^{s} \frac{K}{p} \right] - f(\vec{x}) & s \neq p \\ 0 & otherwise \end{cases}$$

where K is a large constant [O(10^9)], p is the number of constraints and s is the number of these which have been satisfied. K's only restriction is that it should be large enough to insure that any non-feasible individual is graded much more poorly than any feasible one. Here the algorithm receives information as to how many constraints have been satisfied but is not otherwise affected by the strategy. Furthermore, to solve the problem we used a fixed point format with I bits for the integer and D bits for the decimal plus one sign bit. Therefore, the size of the chromosome (κ) is

$$\kappa = | \; Chromosome \; | = (I + D + 1) \times Number \;\; of \;\; variables \tag{7}$$

With these specifications we set the Genetic Algorithm as follows:

a) Number of individuals → 250
b) Bits for integers → 12
c) Bits for decimals → 32
   $\kappa = (12+32+1) \times 4 = 180$
d) P_c=0.9
e) P_m=0.01
f) Number of generations → 4,000

For these settings the results were as follows:

$$V_1^*(\tau) = -994.42$$
$$V_3^*(\tau)=+108.625$$
$$V_5^*(\tau)=+1{,}861.43$$
$$V_{12}^*(\tau)=+384.9302$$

for which $C(\tau) \approx 89{,}762{,}858.74$. The reader may verify that these values comply with all the restrictions. Furthermore, the value of C(τ), which reflects the monthly cost of operation of the system, lies well under the typical average costs of the actual day to day operation of the system.

6 Conclusions

We have shown that a complex dynamic system may be analyzed from its past behavior and, given a known cost function, the relevant variables for its operation may be calculated in advance in order for the system to perform close to the optimum. To do this we found a set of multivariate functions in the form of Multi-Layered Perceptron Networks. Once these were found, we were able to optimize the forecasted function's values by including an uncertainty factor. This factor has to do with the estimated forecasting error. This consideration led us to consider a constrained optimization problem which we tackled using a non-traditional genetic algorithm which has been cited in the literature as superior to other evolutionary alternatives [11]. We stress that, for this method to work adequately, it is very important to observe a series of pre-processing steps. In our case these were: a) Elimination of highly correlated variables, b) Scaling the values of the variables to make them scale independent, c) Optimal interpolation of the known values to enhance the data sample, d) Proper selection of the network's architecture, e) Proper selection of the error measure. Once all these issues have been dealt with, we proceeded to split the data in two samples in order for us to cross-validate the NNs after the training phase. Finally, an agile treatment of the constraints is needed for the GA to find values closer to the optimum.

This is a prime example of computational intelligence put to work; where classical statistical methods are disallowed because of the high number of variables. Likewise, classical approximation methods rarely will be useful in practice because of the inherent high order of the closed approximants and the related numerical instability [12], [13] which often crops up when solving large systems of equations.

Given all of the above considerations, we think that reliable forecasting and optimization is possible given that the data under study do not exhibit chaotic behavior. In the latter case no forecasting algorithm will perform properly. However, for a well behaved problem, such as the one discussed here, we may extract the patterns embedded in the data to our advantage and are able to optimize such forecasted patterns.

References

1. Haykin, S.: Neural Networks. A comprehensive foundation, 2nd edn. Prentice Hall (1999)
2. Back, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)
3. MIT Staff, Machine Monitoring and Diagnosis using Knowledge-Based Fuzzy Systems. Data Engine Tutorials and Theory, Management Intelligenter Technologien GmbH, Aachen, Germany (2010)
4. Shampine, L., Allen, R.: Numerical Computing: an introduction, pp. 43–53. W.B. Saunders (1984)
5. Rumelhart, D.E., Hintoen, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) PDP Research group, Parallel Distributed Processing. Foundations, vol. I. MIT Press (1986)
6. Montana, D.J., Davis, L.D.: Training feedforward Networks using Genetic Algorithms. In: Proceedings of the International Joint Conference on Artificial Intelligence. Morgan Kauffman (1989)
7. Kuri, A.: A universal Eclectic Genetic Algorithm for Constrained Optimization. In: Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT 1998, pp. 518–522 (1998)
8. Yao, X.: A Review of Evolutionary Artificial Neural Networks. International Journal of Intelligent Systems 8, 539–567 (1993)
9. Coello, C.: Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. Computer Methods in Applied Mechanics and Engineering (2001)
10. Kuri-Morales, Á.F., Gutiérrez-García, J.O.: Penalty Function Methods for Constrained Optimization with Genetic Algorithms: A Statistical Analysis. In: Coello Coello, C.A., de Albornoz, Á., Sucar, L.E., Battistutti, O.C. (eds.) MICAI 2002. LNCS (LNAI), vol. 2313, pp. 108–117. Springer, Heidelberg (2002)
11. Kuri-Morales, Á.F.: A Methodology for the Statistical Characterization of Genetic Algorithms. In: Coello Coello, C.A., de Albornoz, Á., Sucar, L.E., Battistutti, O.C. (eds.) MICAI 2002. LNCS (LNAI), vol. 2313, pp. 79–89. Springer, Heidelberg (2002)
12. Scheid, F.: Theory and Problems of Numerical Analysis, 7th edn. McGraw-Hill Book Company (1997)
13. Cheney, E.W.: Introduction to Approximation Theory, ch. 2, pp. 58–64. McGraw-Hill Book Company (1966)
14. Tryba, V., Kiziloglu, B., Thimm, A., Daehn, W.: Bestimmung der Abwasserqualität mit einem Multilayerperceptron-Netz für die Online-Steuerung von Regenüberlaufbecken. In: Anwendersymposium Erlangen 1996, pp. S131–S137. MIT GmbH (1996)

Recurrent Neural Identification and I-Term Sliding Mode Control of a Vehicle System Using Levenberg-Marquardt Learning

Ieroham Baruch, Sergio-Miguel Hernandez-Manzano, and Jacob Moreno-Cruz

Department of Automatic Control, CINVESTAV-IPN, Ave. IPN No 2508,
A.P. 14-740, 07360 Mexico City, Mexico
baruch@ctrl.cinvestav.mx, manzano_appleton@hotmail.com,
jacob_moreno@yahoo.com.mx

Abstract. A new Modular Recurrent Trainable Neural Network (MRTNN) has been used for system identification of a vehicle motor system. The first MRTNN module identified the exponential part of the unknown vehicle motor plant and the second one - the oscillatory part of that plant. The vehicle motor plant has been controlled by an indirect sliding mode adaptive control system with integral term. The sliding mode controller used the estimated parameters and states to suppress the vehicle plant oscillations and the static plant output control error is reduced by an I-term added to the control.

Keywords: Modular recurrent neural network, indirect adaptive sliding mode control with integral term, vehicle motor nonlinear oscillatory plant, recurrent neural network system identification.

1 Introduction

The Recurrent Neural Networks (RNN) are more powerful computational models then the classical Feedforward Neural Networks (FFNN) due to its ability to process temporal sequences memorizing and remembering its history thanks to its state space representation, [1].

The oscillations are important phenomena in various human activities like science, technology and industry, [2]. Sometimes oscillations are destructive and dangerous, in other occasions they could be just noise or vibrations. In many cases when the dynamics of the system is nonlinear it is preferable to use RNN to identify the oscillatory mechanical plant, [3]. In [4] it is compared the use of RNN with the use of an observer to find the torque applied to the plant. In [5] it is proposed to use a RNN and a PI controller so to suppress vibrations of mechanical system driven by linear AC motor. In [6] it is proposed to use a FFNN identifier and a FFNN controller to suppress system sounds and vibrations. In [7] it is applied a RNN for inverse system modeling and feedforward control of mechanical plant.

In some early publications the authors applied RNNs for system identification and I-term control of mechanical plants, [8]. Here the applied RNN will be extended to

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 304–316, 2013.

modular RNN and learned by the Levenberg-Marquardt (L-M) algorithm so to identify better the studied vehicle plant oscillations. The identified oscillations will be suppressed by indirect sliding mode control system with I-term. The vehicle motor system model, taken from the literature, [9]-[13], will serve as an input/output data generator so to obtain simulation results.

1.1 Mathematical Model of the Vehicle Motor Plant

The modern vehicle motor control systems represent Electronic Control Units (ECU) controlling the Idle Speed (IS) and the ignition of the motor, [9]-[13]. The mathematical model of the vehicle motor (volume of 1.6 litters, 4 cylinders) is given by the following system of nonlinear differential equations:

$$\dot{P} = k_P\left(\dot{m}_{ai} - \dot{m}_{ao}\right) \tag{1}$$

$$\dot{N} = k_N\left(T_i - T_L\right) \tag{2}$$

$$\dot{m}_{ai} = \left(1 + 0.907\theta + 0.0998\theta^2\right)g(P) \tag{3}$$

$$g(P) = \begin{cases} 1 & P < 50.6625 \\ 0.0197\left(101.325P - P^2\right)^{1/2} & P >= 50.6625 \end{cases} \tag{4}$$

$$\dot{m}_{ao} = -0.0005968\ N - 0.1336\,P + 0.0005341\ NP + 0.000001757\ NP^2 \tag{5}$$

$$m_{ao} = \dot{m}_{ao}\left(t - \tau\right)/\left(120\ N\right) \tag{6}$$

$$\tau = \frac{45}{N} \tag{7}$$

$$\begin{aligned} T_i = {} & -39.22 + 325024\,m_{ao} - 0.0112\,\delta^2 + 0.635\,\delta + \left(0.0216 + 0.000675\,\delta\right) \\ & N\left(2\pi/60\right) - 0.000102\ N^2\left(2\pi/60\right)^2 \end{aligned} \tag{8}$$

$$T_L = \left(N/263.17\right)^2 + T_d \tag{9}$$

Where: P is the inside gazes motor pressure (measured in kPa); N is the motor velocity (rpm); θ (changed in interval 5-25) and δ (changed in interval 10-45) are control signals (accelerator angle and ignition angle measured in grads); Td represent the outside perturbation torques (changed in the interval 0-61 and measured in N-m); m_{ai} and m_{ao} are input and output (cylinder) air masses; Ti is the motor torque; T_L is the load torque; τ is a time delay; g(P) is a nonlinear function. The vehicle motor plant outputs are the pressure P and the velocity N. This nonlinear oscillatory plant will be identified by Modular Recurrent Trainable Neural Network (MRTNN).

2 Topology and BP Learning of the Modular RTNN

The modular RTNN topology is depicted in Fig. 1. The first module represents the real (exponential) part of the RTNN. The second module represents the complex (oscillatory) part of the RTNN. Using the diagrammatic rules, [14], we could derive and depict the adjoint RTNN (see Fig. 2) used for learning.

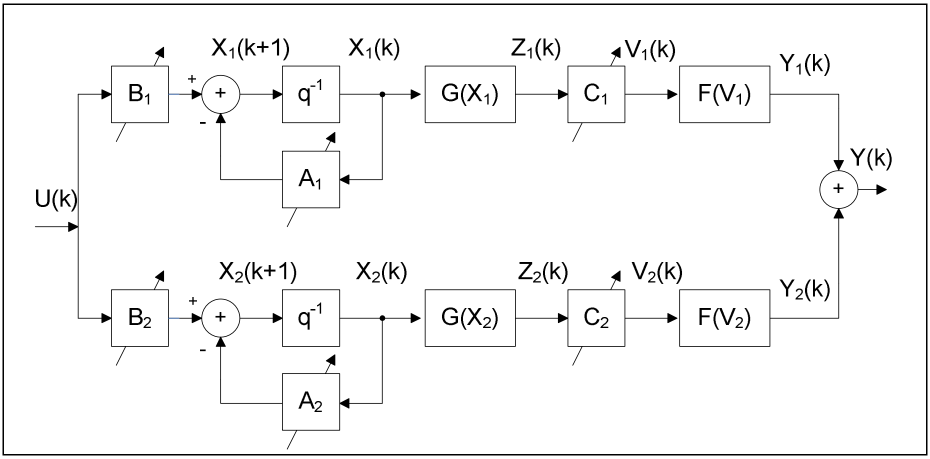


Fig. 1. Topology of the modular RTNN

Following Figs. 1, 2, we could derive the dynamic BP algorithm of its learning based on the Modular Recurrent Trainable NN (MRTNN) topology, using the diagrammatic method, [14]. The modular RTNN topology and its BP learning algorithm, [14], are described in vector-matrix form as:

$$X(k+1) = AX(k) + BU(k);\ Z(k) = G[X(k)]; \tag{10}$$

$$V(k) = CZ(k);\ Y(k) = F[V(k)];\ X = [X_1\ ;\ X_2]^T\ ;\ Z = [Z_1\ ;\ Z_2]^T;\ V = [V_1\ ;\ V_2]^T \tag{11}$$

$$Y = [Y_1\ ;\ Y_2]^T\ ;\ B = [B_1\ ;\ B_2]^T;\ C = [C_1\ ;\ C_2]; \tag{12}$$

$$A = \text{block-diag}\ (Ai),\ i=1,2; \tag{13}$$

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}; A_1 = diag(\lambda_1); |\ \lambda_1\ | < 1; \tag{14}$$

$$A_2 = diag(A_{2i}); \mathrm{A}_{2i} = \begin{bmatrix} a_{1i} & a_{2i} \\ a_{3i} & a_{4i} \end{bmatrix}; \tag{15}$$
$$\alpha = tr(A_{2i})/2 < 1; \beta = \det(A_{2i}) < 1; (\alpha - \beta) < 0$$

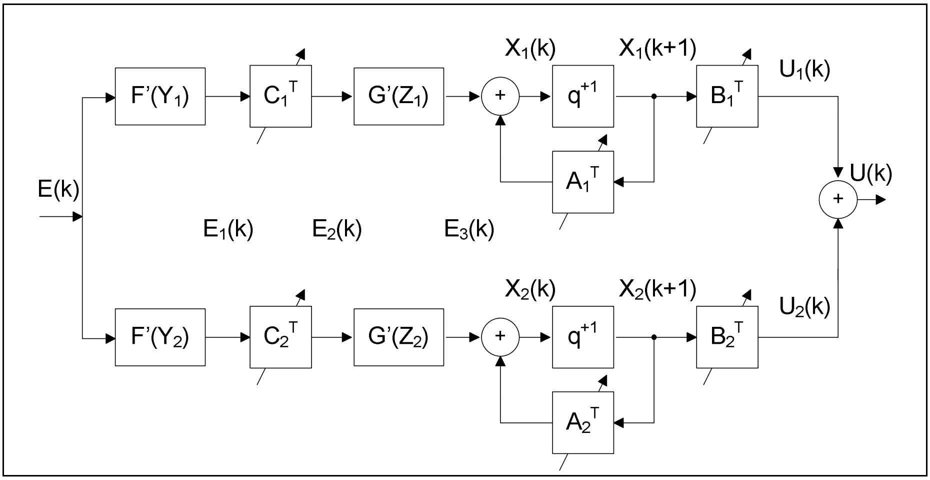


Fig. 2. Adjoint topology of the modular RTNN

$$W(k+1) = W(k) + \eta\, \Delta W(k) + \alpha\, \Delta Wij(k-1); \tag{16}$$

$$E(k) = T(k)-Y(k);\ E_1(k) = F'[Y(k)]\, E(k);\ F'[Y(k)] = [1-Y_2(k)]; \tag{17}$$

$$E_3(k) = G'[Z(k)]\, E_2(k);\ E_2(k) = C^T(k)\, E_1(k);\ G'[Z(k)] = [1-Z_2(k)]; \tag{18}$$

$$\Delta C(k) = E_1(k)\, Z^T(k);\ \Delta B(k) = E_3(k)\, U^T(k); \tag{19}$$

$$\Delta A_1(k) = E_{31}(k)\ {}^{\circ}X_1(k);\ \Delta A_2(k) = E_{32}(k)\, X_2{}^T(k) \tag{20}$$

$$E = [E_1\ ;\ E_2]^T\ ;\ E_1 = [E_{11}\ ;\ E_{12}]^T;\ E_2 = [E_{21}\ ;\ E_{22}]^T;\ E_3 = [E_{31}\ ;\ E_{32}]^T \tag{21}$$

where: X, Y, U are state, output, and input vectors with dimensions n, l, m, respectively; Z is n-dimensional output of the hidden layer; V is l-dimensional post-synaptic activity of the output layer; T is l-dimensional plant output vector, considered as a RTNN reference; A is (nxn) state weight matrix; B and C are (nxm) and (lxn) input and output weight matrices; F[.], G[.] are vector-valued tanh(.)-activation functions; F'[.], G'[.] are the derivatives of these functions; the state dimensions of both exponential and oscillatory RTNN modules (see Fig.1) are n_1 and n_2 (this dimension is even and n= n_1+n_2), l_1 and l_2 (l= l_1+l_2) are output dimensions of the exponential and oscillatory modules of the modular RTNN. The matrix W is a general weight, denoting each weight matrix (C, A, B) to be updated; ΔW (ΔC, ΔA, ΔB), is the weight correction of W; η, α are learning rate parameters: The diagonal blocks of the state matrix A corresponds to the exponential and the oscillatory modules of the RTNN. The error vectors E, E_1, E_2, E_3, predicted by the adjoint RTNN model, have appropriate dimensions, (see Fig.2). These vectors have also two sub-vectors with appropriate dimensions, corresponding to the output and state dimensions of the exponential and oscillatory RTNN modules. Note that the RTNN topology in fact represented a nonlinear Wiener-Hammerstein model where the states depend linearly on the affine inputs and the outputs of both layers are nonlinear. The controllability and observability (the identifiability) of this RTNN model is discussed in [14]. The stability of the RTNN model is assured by the activation functions (-1, 1) bounds and by the local stability weight bound condition, given by (13)-(15). The first two conditions for the matrix A_2 are stability conditions and the last condition for the matrix A_2 is condition for oscillation existence. The learning of the exponential part of the feedback matrix is an element by element product of two vectors (see (20)).

3 Recursive Levenberg-Marquardt Learning of the MRTNN

The application of the recursive L-M algorithm for learning of MRTNN is the new point of this paper. This algorithm does not need special measures to preserve its stability during the learning like others BP-based algorithm does, [15], [16]. The general recursive L-M algorithm of learning, [14], is given by the next equations:

$$W(k+1)=W(k)+P(k)\nabla Y[W(k)]E[W(k)] \tag{22}$$

$$Y[W(k)]=g[W(k),U(k)] \tag{23}$$

$$E^2[W(k)]=\{Y_p(k)-g[W(k),U(k)]\}^2 \tag{24}$$

$$DY[W(k)]=\frac{\partial}{\partial W}g[W,U(k)]\Big|_{W=W(k)} \tag{25}$$

Where: W is a general weight matrix (A, B, C) under modification; P is a symmetric matrix updated by (28); DY[.] is an nw-dimensional gradient vector; Y is the RTNN output vector which depends of the updated weights and the input; E is an error vector; Yp is the plant output vector, which is in fact the target vector. Using the same RTNN adjoint block diagram (see Fig.2), it is possible to obtain the values of the gradients DY[.] for each updated weight, propagating the value D(k) = I through it. Here the matrices D are sectioned in parts corresponding to the division of the error vectors. Applying equation (25), the corresponding gradient components are obtained as follows:

$$DY[C_{ij}(k)]=D_{1,i}(k)Z_j(k);\; DY[A_{ij}(k)]=D_{2,i}(k)X_j(k);\; DY[B_{ij}(k)]=D_{2,i}(k)U_j(k) \tag{26}$$

$$D_{1,i}(k)=F_i'[Y_i(k)];\; D_{2,i}(k)=G_i'[Z_i(k)]C_iD_{1,i}(k) \tag{27}$$

The P(k) matrix was computed recursively by the equation:

$$P(k)=\alpha^{-1}(k)\{P(k-1)-P(k-1).\Omega[W(k)]S^{-1}[W(k)]\Omega^T[W(k)]P(k-1)\} \tag{28}$$

where the S(.), and Ω(.) matrices were given as follows:

$$S[W(k)] = \alpha(k)\Lambda(k) + \Omega^T[W(k)]P(k-1)\Omega[W(k)] \tag{29}$$

$$\begin{aligned}
&\Omega^T[W(k)] = \begin{bmatrix} & & \nabla Y^T[W(k)] & & \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}; \\
&\Lambda(k)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}; 10^{-4} \le \rho \le 10^{-6}; \\
&0.97 \le \alpha(k) \le 1; 10^3 \le P(0) \le 10^6; \\
&i = k \bmod(nw) + 1; k > nw
\end{aligned} \tag{30}$$

The matrix Ω(.) has dimension (nwx2), whereas the second row has only one unity element (the others were zero). The position of that element was computed by the last statement of (30). After this, a short description of the sliding mode Indirect Adaptive Neural Control (IANC) system with I-term will be given.

4 Indirect (Sliding Mode) Adaptive Control System with I-Term

The block-diagram of the control system is given on Fig.3. It contained a recurrent neural identifier RTNN-1, an I-term, control and a Sliding Mode (SM) controller with entries – the reference signal R, the output error Ec, and the states X and parameters A, B, C, estimated by the neural identifier RTNN1. The total control is a sum of the SM control and the I-term control (with dimension l), computed using the equation:

$$V(k+1) = V(k) + To\ Ec(k) \tag{31}$$

Let us suppose that the studied local nonlinear plant possess the following structure:

$$X_p(k+1)=F[X_p(k),U(k)];\ Y_p(k)=G[X_p(k)] \tag{32}$$

Where: $X_p(k)$, $Y_p(k)$, U(k) are plant state, output and input vector variables with dimensions np, l and m (l = m is supposed); F and G are smooth, odd, bounded nonlinear functions. The linearization of the activation functions of the local learned identification RTNN model, leads to the following linear local plant model:

$$X(k+1)=AX(k)+BU(k);\ Y(k)=CX(k) \tag{33}$$

Let us define the following sliding surface error function S(·) with respect to the output tracking error:

$$S(k+1) = E(k+1) + \sum_{i=1}^{P} \gamma_i E(k-i+1)\ ;\ \ |\gamma_i| < 1; \tag{34}$$

Where: E(·) is the systems local output tracking error; γ_i are parameters of the local desired error function; p is the order of the error function. The local tracking error is defined as E(k)=R(k)-Y(k), where: R(k) is a l-dimensional local reference vector.

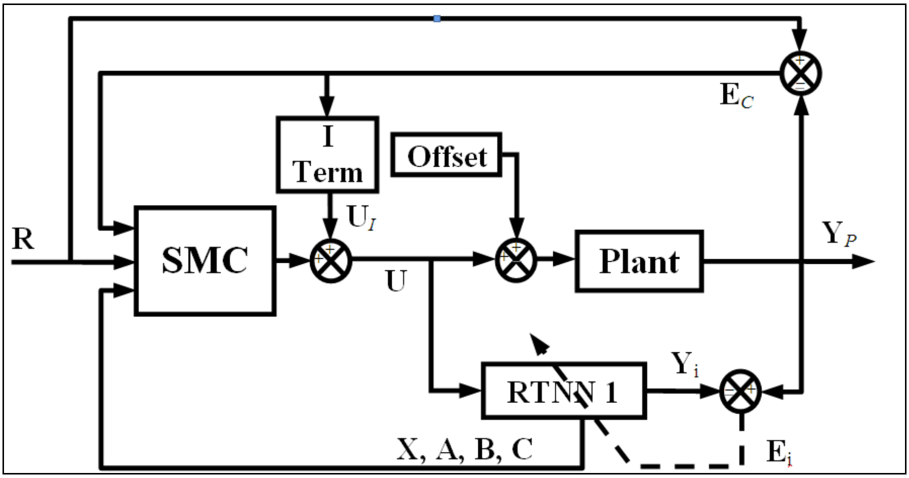


Fig. 3. Block diagram of the indirect adaptive SM control with I-term

The objective of the sliding mode control systems design is to find a control action which maintains the systems error on the sliding surface assuring that the output tracking error reached zero in p steps, where p<n, which is fulfilled if S(k+1)=0. As the local approximation plant model (33), is controllable, observable and stable, [14], the matrix A is block-diagonal, and l=m, the matrix product (CB) is nonsingular, and the plant states X(k) are smooth non- increasing functions. Now, from (33), (34), taking into account the mentioned above observations, it is easy to obtain the equivalent control capable to lead the system to the sliding surface:

$$U_{eq}(k) = (CB)^{-1}\left[-CAX(k) + R(k+1) + \sum_{i=1}^{P} \gamma_i E(k-i+1)\right] \tag{35}$$

The SMC avoiding chattering is taken using a saturation function inside a bounded control level Uo (taking into account plant uncertainties). The proposed SMC cope with the characteristics of the wide class of plant model reduction neural control with reference model, and represents an indirect adaptive neural control, [14].

5 Simulation Results

The system Identification was performed by a Modular RTNN. The topology of the Modular RTNN is (2, 5, 2), where one block with state dimension 1 and two blocks with state dimension 2 are used. The activation functions are tanh(.) for both layers of the RTNN. The learning rate parameter of L-M is $\rho = 4$. The plant model, used as an input/output data generator is given by equations (1)-(9). The simulation results of MRTNN system identification are obtained on-line during 2.5 seconds using both BP and L-M learning. Both identification inputs are combination of tree sinusoids. The graphical simulation results obtained by L-M learning are given on Fig. 4. The quality of learning has been studied using some statistics obtained repeating 20 times the learning experiment and computing the final value of the Means Squared Error (MSE) of learning. The statistical values are computed by the following formulas:

$$\varepsilon = \frac{1}{n}\sum_{k=1}^{n}\xi_{avk}\,;\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Delta_i^2}\,;\Delta = \xi_{av} - \varepsilon \qquad (36)$$

Where: ε is the mean value of the MSE; σ is the standard deviation; Δ is the deviation with respect to the mean. The final MSE for 20 runs of the L-M and BP learning algorithms are given on Table 1and Table 2, respectively. The Table 3 gives the values of ε and σ obtained by BP and L-M learning algorithms.

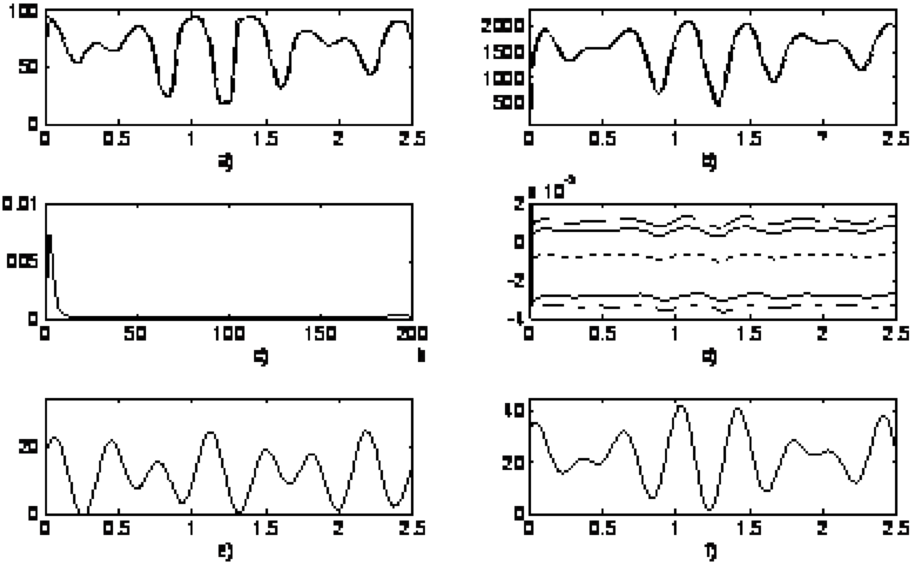


Fig. 4. Graphical simulation results of RTNN system identification obtained by L-M learning algorithm; Continued line-plant output, Pointed line-RTNN output; a) Pressure; b) Velocity; c) MSE; d) Estimated RTNN states; e),f) First and Second plant inputs

Table 1. Final MSE% of 20 runs of the identification program using the RTNN L-M algorithm

No	1	2	3	4	5
MSE%	1,05E-06	9,82E-07	1,03E-06	1,02E-06	1,04E-06
No	6	7	8	9	10
MSE%	1,00E-06	9,90E-07	1,05E-06	1,05E-06	9,80E-07
No	11	12	13	14	15
MSE%	1,07E-06	1,01E-06	1,05E-06	1,07E-06	1,00E-06
No	16	17	18	19	20
MSE%	1,06E-06	9,74E-07	1,01E-06	1,02E-06	9,93E-07

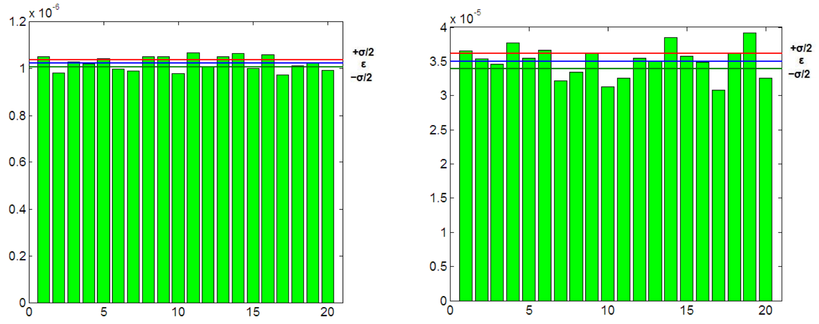
Table 2. Final MSE% of 20 runs of the identification program using the RTNN BP algorithm

No	1	2	3	4	5
MSE%	0.36E-04	3,54E-05	3,46E-05	3,78E-05	3,54E-05
No	6	7	8	9	10
MSE%	3,67E-05	3,22E-05	3,35E-05	3,63E-05	3,13E-05
No	11	12	13	14	15
MSE%	3,26E-05	3,55E-05	3,50E-05	3,85E-05	3,58E-05
No	16	17	18	19	20
MSE%	3,49E-05	3,08E-05	3,62E-05	3,91E-05	3,25E-05

Table 3. Standard deviations and mean average values of identification validation using the BP and L-M algorithms of RTNN learning

BP algorithm	**L-M algorithm**
ε = 8.6712E-06	ε = 3.11E-06
σ = 3.0235E-08	σ = 1.38E-8

The numerical results given in Tables 2, 3, 4 are illustrated by Figures 5 a, b.



a) b)

Fig. 5. Comparison between the final MSE for 20 runs of the identification program: a) using L-M algorithm of learning, b) using BP algorithm of learning

The comparative results showed inferior MSE, ε, and σ for the L-M algorithm of learning with respect to the BP one.

The real-time I Term IANC (see Fig.3) contained M-RTNN1 identifier, as used before, SM controller, and I-term. The M-RTNN was learned by the L-M algorithm with topology (2, 5, 2). The simulation results of IANC are obtained on-line during 10,000 iterations. The Figure 6 showed the control results, for the total time and shorter time of learning, respectively, the MSE of the control system and the MSE of closed loop identification. The final MSE is given in Table 3 for 20 runs of the control program. The Fig.7 illustrate the Table 3 results and the computed statistics which are ε= 4.0768E-06, and σ=6.5208E-08. For sake of comparison, the Fig. 8 gives the simulation results of proportional IANC (without I-term). The results show discrepancy between the system outputs and its corresponding references.

The comparison between the graphical results of Fig,6 and Fig.8 show that the constant additive 10% input noise is completely eliminated by the I-term but it still remain in the case that we withdraw the I-term from the control. The obtained 20 run bar-graphs given on Fig.7 showed a good working of the L-M learning and the sliding mode control, where the computed mead and variance values are low.

Table 4. Final MSE% of 20 runs of I-term IANC program using the RTNN L-M algorithm

No	1	2	3	4	5
MSE%	4.12E-6	3.99E-06	3.95E-06	4.12-E06	4.10E-06
No	6	7	8	9	10
MSE%	4.11E-06	4.09E-06	4.04E-06	4.03E-06	4.14E-06
No	11	12	13	14	15
MSE%	4.01E-06	4.13E-06	4.13E-06	4.14E-06	4.06E-06
No	16	17	18	19	20
MSE%	4.09E-06	4.17E-06	4.04-06	3.95E-06	4.15E-06

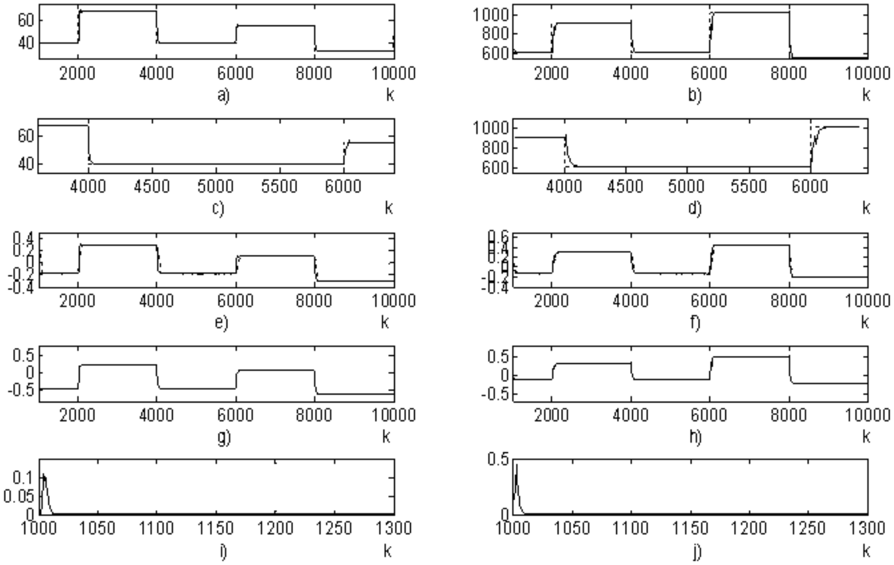


Fig. 6. Graphical simulation results of I-term IANC using L-M learning algorithm; Continued line-plant output, Pointed line-system reference; a) Pressure (total time); b) Velocity (total time); c) Pressure (partial time); d) Velocity (partial time); Plant output vs. MRTNN output; e) Pressure (total time) ; f) Velocity (total time); g) First control; h) Second control; i) MSE of identification; j) MSE of control

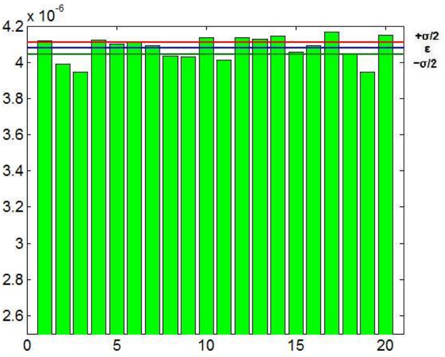


Fig. 7. Comparison between the final MSE for 20 runs of the I-term IANC program using L-M algorithm of learning

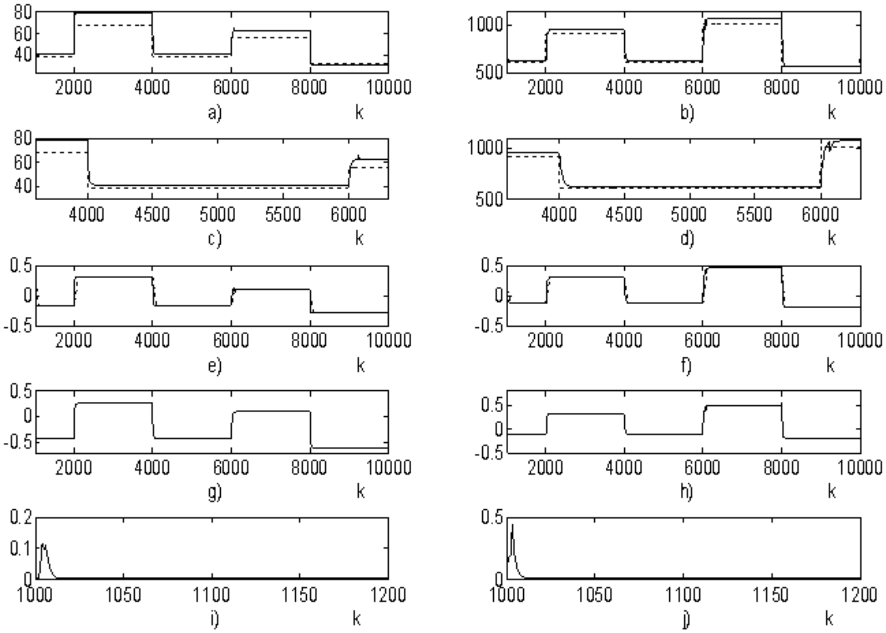


Fig. 8. Graphical simulation results of proportional IANC using L-M learning algorithm; Continued line-plant output, Pointed line-system reference; a) Pressure (total time); b) Velocity (total time); c) Pressure (partial time); d) Velocity (partial time); Plant outputs vs. MRTNN outputs; e) Pressure (total time) ; f) Velocity (total time); g) First control; h) Second control; i) MSE of identification; j) MSE of control.

6 Conclusions

The paper proposed to use a new MRTNN for system identification and indirect (sliding mode) adaptive neural control of a vehicle motor plant. The first MRTNN module identified the exponential part of the unknown plant and the second one- the oscillatory part of the plant. The sliding mode controller used the estimated states and parameters to suppress the plant oscillations. The plant noise is eliminated by an I-term added to the system. The good quality of the proposed IANC system is confirmed by simulation results obtained with a vehicle motor oscillatory plant which mathematical model is taken from the literature. The graphical and numerical (MSE) results show a fast L-M convergence, and great IANC precision.

Acknowledgments. The MS student Jacob Moreno Cruz and the Ph.D. student Sergio-Miguel Hernandez Manzano are thankful to CONACYT, Mexico for the scholarships received during their studies in the Department of Automatic Control, CINVESTAV-IPN, Mexico City, MEXICO.

References

1. Dreyfus, G.: Neural Networks: Methodology and Applications, 2nd edn. Springer, Heidelberg (2005)
2. Fidlin, A.: Nonlinear Oscillations in Mechanical Engineering, 2nd edn. Springer, Heidelberg (2006)
3. Calderon, G., Draye, J.: Nonlinear Dynamic System Identification with Dynamic Recurrent Neural Networks. In: Proc. of the International Workshop on Neural Networks for Identification, Control, Robotics and Signal/Image Processing, pp. 49–54 (1996) ISBN: 0-8186-7456-3
4. Song, J., Lee, K., Choi, J.: Vibration Control of Two Mass System Using a Neural Network Torsional Torque Estimator. In: Proc. of the 24-th Annual Conference of the IEEE, IECON 1998, vol. 3, pp. 1785–1788 (1998) ISBN: 0-7803-4503
5. Yousefi, H., Hirvonen, M., Handroos, H., Soleymani, A.: Application of Neural Network in Suppressing Mechanical Vibration of a Permanent Magnet Linear Motor. Control Engineering Practice 16, 787–797 (2007), doi:10.1016/j.conengprac
6. Bouchard, M.: New Recursive Least-Squares Algorithms for Nonlinear Active Control of Sound and Vibration Using Neural Networks. IEEE Transactions on Neural Networks 12(1), 135–147 (2001)
7. Luis, G., Teixeira, R., Ribero, J.: A Neural Network Based Direct Inverse Control for Active Control of Vibrations of Mechanical Systems. In: Proc. of the IEEE VI Brazilian Symposium on Neural Networks, pp. 107–112 (January 2000) ISBN: 0-7695-0856-1
8. Baruch, I.S., Garrido, R.: A Direct Adaptive Neural Control Scheme with Integral Terms. International Journal of Intelligent Systems, Special issue on Soft Computing for Modelling, Simulation and Control of Nonlinear Dynamical Systems 20(2), 213–224 (2005) ISSN 0884-8173
9. Marko, K.: Neural Network Application to Diagnostics and Control of Vehicle Control Systems. In: Lippmann, R., Moody, J.E., Touretzky, D.S. (eds.) Advances in Neural Information Processing Systems, vol. 3, pp. 537–543. Morgan Kaufmann, San Francisco (1991)

10. Bloch, G., Lauer, F., Colin, G.: On Learning Machines for Engine Control. SCI, pp. 165–189. Springer, Heidelberg (2008)
11. Puskorius, G., Feldkamp, L.: Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks. IEEE Transactions on Neural Networks 5(2), 279–297 (1994) ISSN: 0018-9219
12. Ayeb, M., Lichtenthaler, D., Winsel, T., Theuerkauf, J.: SI Engine Modeling Using Neural Networks. SAE technical paper series, Electronic Engine Controls. Diganostics and Controls (1998), doi:10.4271/980790
13. Kara -Togunm, N., Baysec, S.: rediction of Torque and Specific Fuel Consumption of a Gasoline Engine by Using Artificial Neural Networks. Applied Energy (September 2009), doi:10.1016/j.apenergy.2009.08.016
14. Baruch, I.S., Mariaca-Gaspar, C.R.: A Levenberg-Marquardt Learning Applied for Recurrent Neural Identification and Control of a Wastewater Treatment Bioprocess. International Journal of Intelligent Systems 24, 1094–1114 (2009) ISSN 0884-8173
15. Kazemy, A., Hosseini, S.A., Farrokhi, M.: Second Order Diagonal Recurrent Neural Network. In: IEEE International Symposium on Industrial Electronics, ISIE, pp. 251–256 (2007), doi:10.1109/isie.2007.4374607
16. Govindhasamy, J.J., McLoone, S.F., Irvin, G.W.: Second Order Training of Adaptive Critics for Online Process Control. IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics 35(2), 381–385 (2005), doi:10.1109/TSMCB.2004.843276

Multiple Fault Diagnosis in Electrical Power Systems with Dynamic Load Changes Using Soft Computing

Juan Pablo Nieto González

Corporación Mexicana de Investigación en Materiales,
COMIMSA S.A. de C.V.
Ciencia y Tecnología No. 790, Fracc. Saltillo 400, C.P. 25290
Saltillo, Coahuila, México
juan.nieto@comimsa.com

Abstract. Power systems monitoring is particularly challenging due to the presence of dynamic load changes in normal operation mode of network nodes, as well as the presence of both continuous and discrete variables, noisy information and lack or excess of data. In this domain, the need to develop more powerful approaches has been recognized, and hybrid techniques that combine several reasoning methods start to be used. The present work is a variant of a proposal made by the author. This paper proposes a fault diagnosis framework that is able to locate the set of nodes involved in multiple fault events. The proposal is a methodology based on the system history data. It detects the faulty nodes, the type of fault in those nodes and the time when it is present. The framework is composed of two phases: In the first phase a probabilistic neural network is trained with the eigenvalues of voltage data collected during normal operation, symmetrical and asymmetrical fault disturbances. The second phase uses an Adaptive Neuro-Fuzzy Inference Systems (ANFIS) to give the final diagnosis. A set of simulations are carried out over an electrical power system proposed by the IEEE. To show the performance of the approach, a comparison is made against two different diagnostic systems.

Keywords: Fault Detection, Fault Diagnosis, Electrical Power System, Dynamic Load Changes, Soft Computing, Probabilistic Neural Network, Eigenvalues, ANFIS.

1 Introduction

From the point of view of safety and reliability of electric power systems, it is necessary to have an early fault diagnosis scheme which can detect, isolate and diagnose the faults, and advise the system operators to initiate corrective action(s). During a disturbance, there is a great number of events related to the fault(s), making its origin acknowledge (diagnosis) and the decision of the restoration actions to be carried out a difficult task to the power system operator. Moreover, with advances in power system devices and communications, even

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 317–328, 2013.

more information will be presented to operators, and alarm processing in large power systems calls for the treatment of a great bulk of information. The increase in information will enable a more complete view of the power systems state, but it will increase the need for fault diagnosis to effectively handle the system information. In this domain, the need to develop more powerful approaches has been recognized, and hybrid techniques that combine several reasoning methods start to be used.

An electric power systems fault detection and diagnosis using the eigenvalues of the correlation matrix, a probabilistic neural network (PNN) and an Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is proposed. This system is a process history base method. The organization of the paper is as follows. Section 2 reviews the state of the art. Section 3 gives the preliminaries and the background knowledge on correlation matrix, eigenvalues, PNN and ANFIS. Section 4 gives the approach general description. Section 5 shows how this framework works in a simulation example. Finally, conclusion ends this paper in section 6.

2 State of the Art

Very recently, the need to develop more powerful approaches has been recognized, and hybrid techniques that combine several reasoning methods start to be used. [1] developed a methodology using wavelet transform for phase to ground fault detection in primary distribution systems, but it is an efficient methodology only for single phase fault detection in unbalanced distribution systems. [2] proposes a fault diagnosis scheme for power distribution systems, composed by three different processes: fault detection and classification, fault location, and fault section determination. The fault detection and classification technique is wavelet based. The fault-location technique is impedance based and uses local voltage and current fundamental phasors. The fault section determination method is artificial neural network based and uses the local current and voltage signals to estimate the faulted section. [3] considers the configuration of automatic devices in modern power systems, such as protective relays and reclosing relays to improve an analytic model and optimization technique-based method for fault diagnosis of power systems. According to the action principle of protective relays, the issue of fault diagnosis is first expressed as a 0-1 integer programming problem and then it is solved by Tabu Search and Genetic algorithm. [4] presents a methodology that uses artificial neural networks integrated with other several statistical techniques.Among the numerical and statistical tools used in the approach is the Fourier parameters, the RMS values (RMS), the constant of false alarm rates (CFAR), the skewness values (SV), the Kurtosis measures (KM), the ratio of power (ROP), symmetrical components (SC), which seek to identify in a integrated way between a normal operation situation and a transient occurrence situation. When there is a fault, an artificial neural network of multilayer perceptron type classifies it. [5] uses readings of the phase current only during the first one-forth of a cycle in an integrated method that combines symmetrical components technique with the principal component analysis (PCA) to declare, identify, and classify a fault. This approach also distinguishes a real fault from a transient one and can be used in either a transmission or a distribution system. [7] proposes an Augmented Naive Bayesian power network fault diagnosis method based on data mining to diagnose faults in power network. The status information of protections and circuit breakers are taken as conditional attributes and faulty region as decision-making attribute. [8] can analyze faults occurring between two buses that are equipped with measurement units. The first step of the framework is to detect the presence of a fault in the power system in real time. Then, the method of symmetrical components is used to convert the three-phase power signals to three sets of independent components, which are positive, negative, and zero sequences. Thus, in this work a new electrical power system fault detection and diagnosis proposal based on the history data using a PNN and an ANFIS is proposed (PNN + ANFIS).

3 Preliminars

3.1 Correlation Matrix and Eigenvalues Definitions

Correlation Matrix Definiton. A Correlation matrix describes correlation among M variables. It is a square symmetrical $M \times M$ matrix with the (ik)th element equal to the correlation coefficient r_{ik} between the (i)th and the (k)th variable. The correlation coefficient is obtained as

$$r_{ik} = \frac{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)(x_{jk} - \bar{x}_k)}{\sqrt{\sum_{j=1}^{n}(x_{ji} - \bar{x}_i)^2}\sqrt{\sum_{j=1}^{n}(x_{jk} - \bar{x}_k)^2}} \tag{1}$$

The diagonal elements (correlations of variables with themselves) are always equal to 1.00 [11].

Eigenvalue Definition. Let A be a $k \times k$ square matrix and I be the $k \times k$ identity matrix. Then the scalars

$$\lambda_1, \lambda_2, ..., \lambda_k \tag{2}$$

satisfying the polynomial equation

$$|A - \lambda I| = 0 \tag{3}$$

are called the eigenvalues or characteristic roots of a matrix A. The equation $|A - \lambda I| = 0$ is called the characteristic equation, thus similar matrices and A and its transpose matrix have same eigenvalues [11].

3.2 Probabilistic Neural Network

Probabilistic Neural Network (PNN) are conceptually similar to K-Nearest Neighbor (KNN) models [10], [12]. The basic idea is that a predicted value of an item is

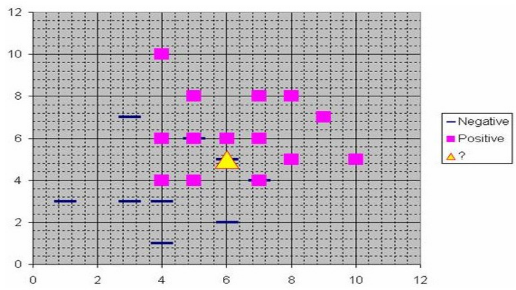


Fig. 1. PNN are conceptually similar to KNN

likely to be about the same as other items that have close values of the predictor variables.

For example, from Fig. 1 it is assumed that each case in the training set has two predictor variables, x and y. The cases are plotted using their x, y coordinates as shown in the figure. Also we assume that the target variable has two categories, positive which is denoted by a square and negative which is denoted by a dash. It can be noted that the triangle is positioned almost exactly on top of a dash representing a negative value. But that dash is in a fairly unusual position compared to the other dashes which are clustered below the squares and left of center. So it could be that the underlying negative value is an odd case. The nearest neighbor classification will depend on how many neighboring points are considered. If 1-NN is used and only the closest point is considered, then the new point should be classified as negative since it is on top of a known negative point. On the other hand, if 9-NN classification is used, the closest 9 points are considered and then the effect of the surrounding 8 positive points may overbalance the close negative point. A probabilistic neural network builds on this foundation and generalizes it to consider all of the other points. The distance is computed from the point being evaluated to each of the other points, and a radial basis function (RBF) (also called a kernel function) is applied to the distance to compute the weight (influence) for each point. $Weight = RBF(distance)$ the further some other point is from the new point, the less influence it has. The PNN architecture is shown in Fig. 2. The model has two layers: radial basis layer and competitive layer.

There are Q input vector/target vector pairs. Each target vector has K elements. One of these elements is 1 and the rest is 0. Thus, each input vector is associated with one of K classes. When an input is presented the $||dist||$ box produces a vector whose elements indicate how close the input is to the vectors of the training set. An input vector close to a training vector is represented by a number close to 1 in the output vector a^1. If an input is close to several training vectors of a single class, it is represented by several elements of a^1 that are close to 1. Each vector has a 1 only in the row associated with that particular class of input, and 0's elsewhere. The multiplication Ta^1 sums the elements of a^1 due to each of the K input classes. Finally, the second layer, produces a 1

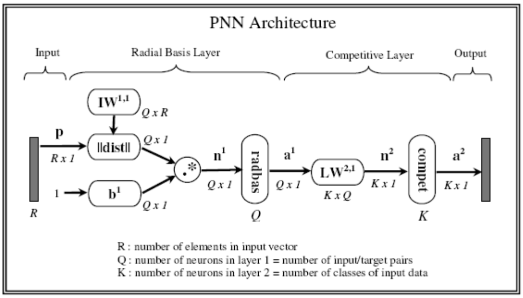

PNN Architecture
Input
Radial Basis Layer
Competitive Layer
Output
IW1,1
Q x R
p
R x 1
||dist||
Q x 1
1
b1
Q x 1
.*
n1
Q x 1
radbas
Q
a1
Q x 1
LW2,1
K x Q
n2
K x 1
compet
K
a2
K x 1
R
R : number of elements in input vector
Q : number of neurons in layer 1 = number of input/target pairs
K : number of neurons in layer 2 = number of classes of input data

Fig. 2. PNN architecture

corresponding to the largest element of n^2, and 0's elsewhere. Thus, the network has classified the input vector into a specific one of K classes because that class had the maximum probability of being correct.

3.3 Adaptive Neuro-Fuzzy Inference Systems

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) are a class of adaptive networks that are functionally equivalent to fuzzy inference systems. For simplicity, assume that the fuzzy inference system under consideration has two inputs x and y and one output z. For a first order Sugeno fuzzy model (shown in Fig. 3) a common rule set with two fuzzy if-then rules is of the form

- Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$
- Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$

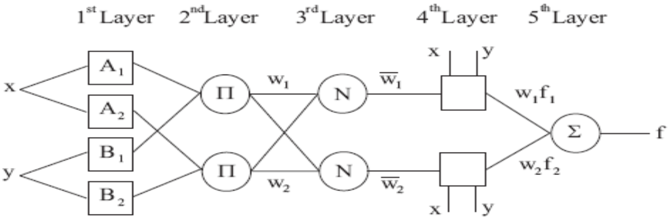


Fig. 3. ANFIS architecture

Every node i in layer 1 has a node function

$$O_{1,i} = \mu_{A_i}(x) \qquad for \;\; i = 1, 2, \tag{4}$$

where $O_{1,i}$ is the membership grade of a fuzzy set A ($A = A_1, A_2, B_1$ or B_2) and it specifies the degree in which the given input x satisfies the quantifier A.

These are the premise parameters. In layer 2 every node is a fixed node labeled Π, whose output is the product of all the incoming signals

$$O_{2,i} = w_i = \mu A_i(x)\mu B_i(y) \qquad i = 1, 2 \tag{5}$$

each node output represents the firing strength of a rule. Every node in layer 3 is a fixed node labeled N. The i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rules' firing strengths

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \qquad i = 1, 2 \tag{6}$$

In layer 4 every node i is an adaptive node with a node function

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{7}$$

these are the consequent parameters. The single node in layer 5 is a fixed node labeled $\sum$, which computes the overall output as the summation of all incoming signals

$$overall\ \ output = O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{8}$$

4 Framework Description

According to [9] this proposal is a process history-based method because of the need of a data set when the system runs under normal operating conditions.

The difference between [6] and this work is the second phase of the proposal. The general detection framework proposed is shown in Fig. 4. The framework only requires a big quantity of historical data, containing normal operation data and different faults present in the system. These data sets are used as prior knowledge to perform the detection process.

The first step is to obtain several data sets from the system. These data sets are matrices formed by windows of m samples and n observed variables. Such matrices are constructed with normal operation and different faults present in system. For each of those data sets its correlation matrix is obtained to see how the observed variables are related. Once having the correlation matrix, their corresponding eigenvalues are computed as shown in subsection 3.1, such that in this way a signature to each of the different possible component states or fault types of the system (K in figure 2) are given. The eigenvalues of the correlation matrix for each fault signature, then serve as the training vectors of the PNN corresponding to Q as described in subsection 3.1.

Fault diagnosis process is carried out in two phases.

1. First phase, that serves as a first filter or information discriminator. When there is a system to monitor, a window of m samples and n observed variables is taken and an analysis is performed over these variables that are related separately. m samples of the system's observed variables are taken

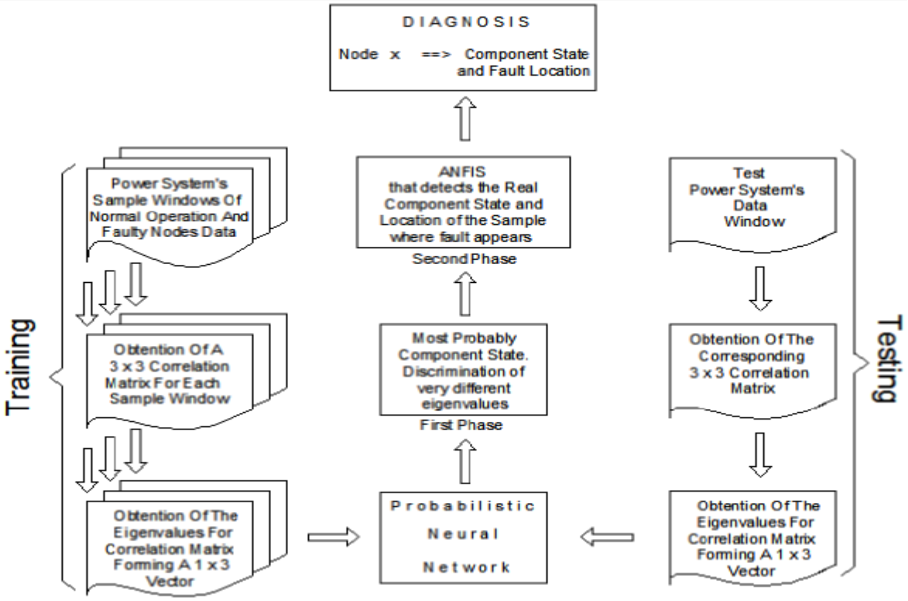

DIAGNOSIS
Node x ==> Component State and Fault Location
ANFIS that detects the Real Component State and Location of the Sample where fault appears
Second Phase
Most Probably Component State. Discrimination of very different eigenvalues
First Phase
Probabilistic Neural Network
Training
Power System's Sample Windows Of Normal Operation And Faulty Nodes Data
Obtention Of A 3 x 3 Correlation Matrix For Each Sample Window
Obtention Of The Eigenvalues For Correlation Matrix Forming A 1 x 3 Vector
Testing
Test Power System's Data Window
Obtention Of The Corresponding 3 x 3 Correlation Matrix
Obtention Of The Eigenvalues For Correlation Matrix Forming A 1 x 3 Vector

Fig. 4. General fault detection and diagnosis framework

to find out which is their most probably state. From these dataset its correlation matrix is obtained. The corresponding eigenvalues of the correlation matrix are obtained and then they are used as an input vector to the PNN previously trained. It is mentioned ”the most probably state” because unfortunately not all the eigenvalues of all of the system states are so different such that PNN could not classify them easily. But it has been found that for certain signature faults eigenvalues are very similar, thus there is here a discrimination/classification phase, because it is necessary to look for the real state but only comparing between just a couple of similar signatures instead of the whole bunch of variables’ states. The output from the PNN automatically discriminates variables’ states that are very different and gives the most probably real related variables’ state.

2. Once the one possible set of related variables’ state is obtained, a second phase of the framework begins to work. In the second phase we build an ANFIS as shown in section 3.3 using for this task according with the most probably state of the system the corresponding variables that are found in normal operation as the predictor variables for each one of the rest of variables. Thus, the ANFIS for a specific variable will predict the value of the variable monitored being this its normal operation value. With this value we obtained a normal operation interval with respect to the normal operation predicted value. Finally a comparison against the normal operation limits

is carried out in order to detect which of the variables has a fault present. This comparison serves as a classifier that gives the real variables' state and can be used to locate the period of time or sample number where the fault occurs.

5 Case Study

The diagnosis system was tailored according to the steps described on section 4. Section 5 shows the performance of the framework proposed for multiple-fault diagnosis over the IEEE network shown in Fig. 5. This is an electrical power system having dynamic load changes. The performance of the methodology proposed for multiple-fault diagnosis was observed within 50 simulation databases of the IEEE network. In order to make the comparison, we have used those 50 databases containing symmetrical and unsymmetrical faults at random nodes, taking into account multiple simultaneous faults scenarios with up to five different faults at a time, and combining faults such as: one line to ground, two lines to ground, three lines to ground, fault between two lines and the no fault mode.

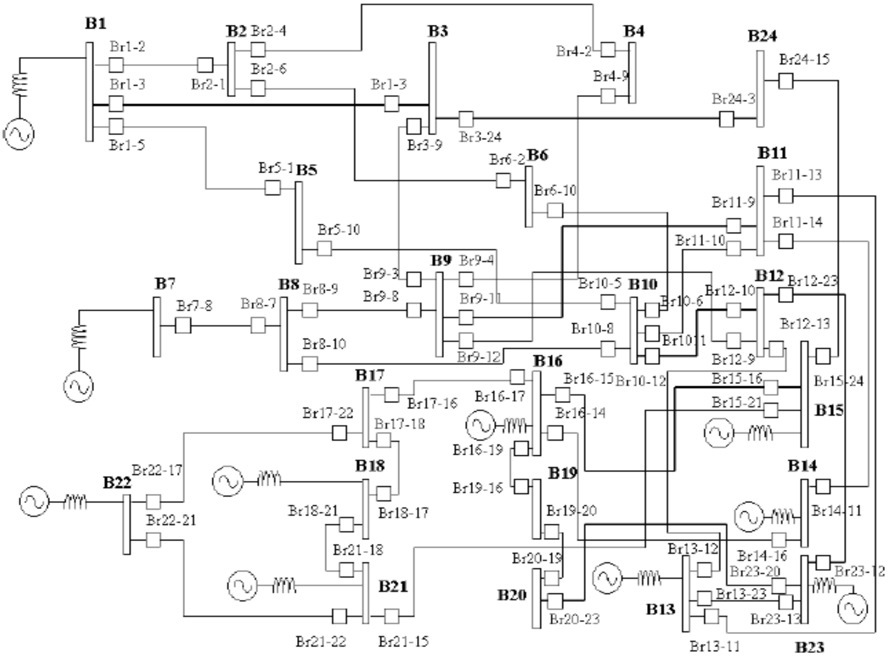


Fig. 5. IEEE reliability test system single line diagram

The methodology proposed is tailored as follows:

1. *Obtain windows of 100 samples from normal and faulty operation history process data (electrical voltage in each node's line).*
2. *Obtain correlation matrix for each node, which gives a* 3×3 *matrix.*
3. *Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), with this 3 eigenvalues form an input vector to train the PNN.*
4. *Take a test data set of 100 samples from the electrical power system being monitored.*
5. *Obtain correlation matrix for each node, which gives a* 3×3 *matrix.*
6. *Obtain the eigenvalues from the correlation matrix (this gives 3 eigenvalues), with this 3 eigenvalues form an input vector to the PNN.*
7. *First Phase: Take the output of the PNN as one of the two most probably states of the node monitored.*
8. *Second Phase: Carry out and ANFIS for each of the variable involved on the most probably state shown by the PNN output. Then compare the suspicious samples with the output of the ANFIS in order to determine the real state of the system and if there is a fault present, classify and locate it.*
9. *Give the diagnosis of each node being monitored. If a fault is present in a specific node give the type and location of it, else print NO FAULT.*

We have considered on the simulations that voltages from the three lines of each of the 24 nodes from the electrical network are measured and registered on a database. We have applied our methodology under the consideration that voltage's information is known, nevertheless on electrical power systems the only available information could be the electrical network's breakers state instead of voltages or electrical current's values.The proposal takes sample windows of 100 data and take into account three possible cases.

- **Case 1:** System is working properly during the first 25 samples from a total of 100, that means 25 samples are ok and 75 samples correspond to fault present on system.
- **Case 2:** Takes 50 samples of normal operation data and 50 samples with fault present.
- **Case 3:** Takes 75 samples of normal operation and 25 with fault present.

Fig. 6 shows the diagnosis of the system using the proposal. The type of fault shown is case 2, one faulty node and two lines to ground. It could be seen how the standardized voltage's magnitude for node 3 plot show that lines A and B go to ground. In the first phase of the method the PNN classifies the eigenvalues of the correlation matrix as a one line to ground fault, but the second phase gives the real type of fault present and it was found that it is a two lines to ground fault. Finally the last plot shows which lines are in a faulty mode according with the result given by the second phase of the method. As expected in this experiment the type of fault is lines A and B to ground.

To observe the performance of the new proposal, a comparison against a diagnostic system based on probabilistic logic taken from [13] and the framework

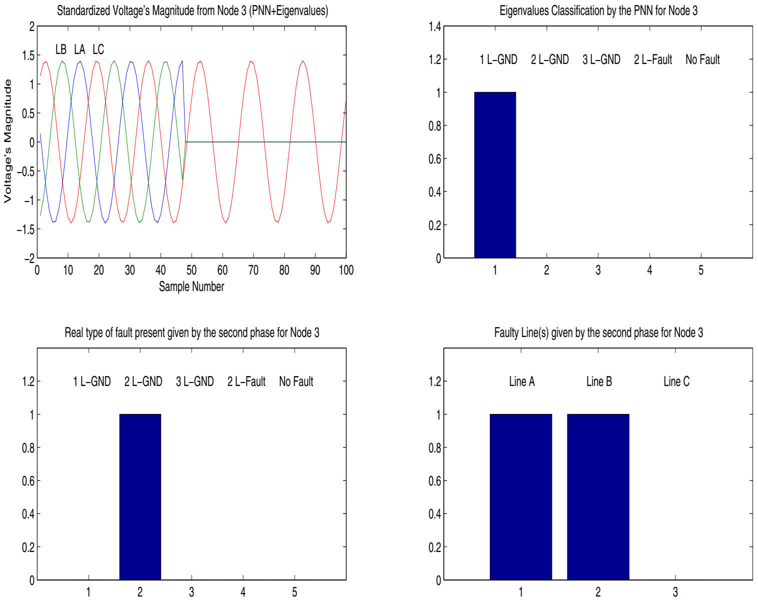


Fig. 6. Diagnosis using PNN and eigenvalues. Fault present on node number 3. The type of fault is case 2, one faulty node and two lines to ground.

Table 1. Comparison of the diagnosis systems general performance's percentages of detection by fault type for each proposal after 50 simulations

Component State	**Probabilistic Logic [13]**	**PNN + EIG [6]**	**PNN + ANFIS**
A-B-C GND	100	100	100
A-B GND	100	100	100
A GND	86	93	100
A-B	83	78	90
B-C	100	79	90
NO FAULT	71	64	85

proposed on [6] has been carried out. Table 1 shows the comparison of the performance's percentages of detection by fault type for each proposal. It clearly shows that the new approach has the best performance of the three diagnostic systems being compared.

Tables 2, 3 and 4 show the comparison made for Case 1, Case 2 and Case 3 respectively. The percentages were obtained after carry out 50 simulations for each case and combining different fault scenarios. First column shows the methods used on each proposal. Second column gives the general percentage of fault detection, that is the percentage of correct detection of a fault present for the 50 simulations. Third column is the general percentage of the correct

Table 2. Comparison of the diagnosis systems general performances percentages for each of the proposals for case 1 after 50 simulations of different fault scenarios

Proposal [Reference]	**Detection**	**Identification**	**Location**
Probabilistic Logic [13]	88	88	70
PNN + EIG [6]	85	85	60
PNN + ANFIS	90	90	90

Table 3. Comparison of the diagnosis systems general performances percentages for each of the proposals for case 2 after 50 simulations of different fault scenarios

Proposal [Reference]	**Detection**	**Identification**	**Location**
Probabilistic Logic [13]	85	83	65
PNN + EIG	79	79	50
PNN + ANFIS	88	88	88

Table 4. Comparison of the diagnosis systems general performances percentages for each of the proposals for case 3 after 50 simulations of different fault scenarios

Proposal [Reference]	**Detection**	**Identification**	**Location**
Probabilistic Logic [13]	80	79	50
PNN + EIG	75	75	45
PNN + ANFIS	85	85	85

identification of type of fault present on the system. Fourth column shows the general percentage of the correct sample location when the fault occurs.

6 Conclusions

This paper has presented a proposal of two phases relatively easy to implement fault diagnosis method. The methodology has been tailored to do a complete fault detection framework for electrical power systems with dynamic load changes. When there is a system to be monitored, it is needed to obtain from it sample windows containing m samples of the voltages presented on each of the three lines forming each power systems nodes. In the first phase, the methodology uses a PNN based on history process data as a first classifier to obtain the most probably operation mode of the nodes being analyzed. Then a second phase is performed in order to give the final diagnosis applying an ANFIS to determine the real state of the variables being monitored in order to diminish the presence of false alarms.

Acknowledgments Author thanks the support received by the CONACYT-FORDECYT project 143332 to do this research work.

References

1. Hartstein, R., Rezende, K., Suman, A.: Fault Detection in Primary Distribution Systems using Wavelets. In: International Conference on Power Systems Transients (ISPT 2007), Lyon, France, pp. 1–6 (2007)
2. Hartstein, R., Rezende, K., Daros, A., Resener, M., Suman, A.: Hybrid Fault Diagnosis Scheme Implementation for Power Distribution System Automation. IEEE Transactions on Power Delivery 23(4), 1846–1856 (2008)
3. Liao, Z., Wen, F., Guo, W., He, X., Jiang, W., Dong, T., Liang, J., Xu, B.: An Analytical Model and Optimization Technique Based Methods for Fault Diagnosis in Power Systems. In: IEEE Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, Nanjing, China, pp. 1388–1393 (April 2008)
4. Flauzino, R.A., Ziolkowski, V., da Silva, I.N., de Souza, D.M.B.: Hybrid Intelligent Architecture for Fault Identification in Power Distribution Systems. In: IEEE Power & Energy Society General Meeting, PES 2009, pp. 1–6 (2009)
5. Alsafasfeh, Q., Abdel-Qader, I., Harb, A.: Symmetrical Pattern and PCA Based Framework for Fault Detection and Classification in Power Systems. In: Conference on Electro Information Technology, pp. 1–5 (2010)
6. Nieto, J.P., Garza, L.E., Morales, R.: Multiple Fault Diagnosis in Electrical Power Systems with Dynamic Load Changes Using Probabilistic Neural Networks. Computación y Sistemas Magazine 14(1), 17–30 (2010)
7. Qianwen, N., Youyuan, W.: An Augmented Naive Bayesian Power Network Fault Diagnosis Method based on Data Mining. In: Asia-Pacific Power and Energy Engineering Conference (APEEC), pp. 1–4 (2011)
8. Jiang, J.A., Chuang, C.L., Wang, Y.C., Hung, C.H., Wang, J.Y., Lee, C.H., Hsiao, Y.T.: A Hybrid Framework for Fault Detection, Classification, and Location Part I: Concept, Structure, and Methodology. IEEE, Transactions on Power Delivery 26(3), 1988–1998 (2011)
9. Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N.: A Review of Process Fault Detection and Diagnosis Part I: Quantitative Model-Based Methods. Computers and Chemical Eng. 27(3), 293–311 (2003)
10. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn (2001) ISBN 0-0471-05669-3
11. Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis, 5th edn. Prentice Hall (2002)
12. Stuart, R., Peter, N.: Artificial Intelligence a Modern Approach, 2nd edn. McGraw-Hill (2003)
13. Garza, L.: Hybrid Systems Fault Diagnosis with a Probabilistic Logic Reasoning Framework. Ph.D. Thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey (2001)

A Hybrid Algorithm for Crustal Velocity Modeling

José Federico Ramírez Cruz[1], Olac Fuentes[2], Rodrigo Romero[2],
and Aaron Velasco[3]

[1] Department of Computer Science,
Instituto Tecnológico de Apizaco,
Av. Tecnológico s/n, Tlaxcala, 90300, México,
[2] Department of Computer Science,
[3] Department of Geological Sciences,
University of Texas at El Paso
500 West University Avenue,
El Paso, TX -79902.
federico_ramirez@yahoo.com.mx,
{fuentes,raromero2,aavelasco}@utep.edu

Abstract. We present a hybrid method to produce a velocity model of the Earth's crust using evolutionary and seismic tomography algorithms. This method takes advantage of the global search ability of an evolution strategy and the quick convergence of an iterative three-dimensional seismic tomography technique to generate a model of the Earth's crustal structure from recorded arrival times of wave fronts produced by controlled sources. The evolution strategy finds a three-dimensional velocity model with constant lateral velocity layers that minimizes the root mean square residuals computed by the tomographic algorithm. The model found is provided as the initial search point to a first arrival traveltime seismic tomography algorithm, which then computes the final three-dimensional velocity model. The method was tested with a real-world data set from an active source experiment performed in the Potrillo Volcanic Field, in Southern New Mexico. Results show that our hybrid method obtains faster convergence and more accurate results than the conventional methods, and does not require an expert-supplied one-dimensional model for the seismic tomography procedure.

1 Introduction

Seismic tomography computes images of the Earth's crustal velocity structure that are used to determine and analyze the internal crustal properties of the Earth. Input data are usually obtained from a set of receivers placed on the Earth surface to record seismic signals generated by passive or active energy sources. Seismic energy can be produced at predetermined locations with controlled explosions, which are referred to as shotpoints and are one type of active source, and it can be sensed by a set of receivers, such as geophones, distributed over the area to survey. Source and receiver locations together with measured

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 329–337, 2013.

first arrival travel times of seismic waves can then be utilized to generate crustal models by combining algorithms for forward modeling, inversion, search, and optimization.

Velocity models obtained with seismic tomography provide information that can be used for a wide variety of applications such as archeological surveys [13], quality control and assessment of engineering projects [11], and discovery of deposits of water, oil, or waste material [10].

Conventional methods for seismic tomography rely on gradient information to guide the search; however, due to noise and the complexity of the search space, multiple local minima commonly exist, thus the accuracy of the final solution depends on the choice of the initial search point, which is normally provided by a human expert [3]. This results in a time-consuming trial-and-error process. Such dependence is typical in standard optimization methods - they need a good initial approximation to find a good solution. With a good initial model, these local optimization algorithms have the advantage of being faster than global methods to reach successful results. In contrast with gradient-based methods, global search methods such as evolutionary algorithms (EAs) can search in very large spaces and are not dependent on the initial solution's proximity to the optimum. EAs have been used to tackle Geophysics inversion problems [5], [12], [14]. In geology, evolutionary algorithms have been used in combination with different techniques [4], [7], [9], yielding satisfactory results.

We designed a hybrid method for seismic tomography that consists of two top-level steps. The first step uses an evolution strategy (ES) to find a one-dimensional (1D) model, or an equivalent flat-layered 3D velocity model, of the surveyed region. The second step uses a first arrival traveltime seismic tomography algorithm, which was developed by Vidale [16] and Hole [8], to compute a 3D velocity model with complex lateral velocity variations. To compute the 1D model, the ES searches for a tridimensional velocity model that minimizes the average root mean square (rms) residuals between first arrival times recorded by receivers and those calculated using the model. The model is discretized at 1km intervals in three dimensions and it assumes constant velocity within each depth layer. The evolution strategy searches for a set of ordered pairs that determines the inflexion points of the 1D model. The remaining values of the model are calculated using linear interpolation. Running times are short because of the low dimensionality of the search problem and the fact that the 1D models obtained after a few generations are accurate enough to be used as starting point to ensure that the 3D gradient-based search will reach a near-optimal solution.

To evaluate the presented method we used data obtained from the Potrillo volcanic field experiment [1]. Test volume dimensions for the experiment are 231 km in width, 26 km in length, and 69 km in depth. The volume was discretized in intervals of 1 km, which produced a 3D grid with 414,414 vertices. The main goal of the method is to compute velocity values for each vertex of the test volume using as inputs the measurements of first arrival travel times of seismic waves generated by 7 shot points and recorded by 793 geophones, the locations of sources and receivers, a velocity range, and a monotonicity constraint on vertex velocities to ensure that velocities increase with depth in the test volume. The large number of parameters is the main challenge to solve this problem.

The next section briefly describes the fundamentals of ES algorithms. Section 3 describes how we used ES to find the initial crustal velocity model for the seismic tomography algorithm. Section 4 presents an analysis and some generated 1D models. The last section presents conclusions and future work.

2 Evolution Strategies

Evolution Strategies (ES) [15,2] are a class of probabilistic search algorithms loosely based on biological evolution that have been applied successfully to optimization problems in poorly-modeled domains and in the presence of noisy data [6]. ES are based on processing a population of individuals. An individual is represented by a vector of real numbers, which is a well-suited representation for problems dealing with continuous parameters. Each vector element is referred to as an object variable x_i and it has an associated standard deviation σ_i, which is referred to as the strategy parameter.

The ES algorithm starts by randomly generating a population of individuals; each individual represents a search point in the space of potential solutions of the problem of interest. Subsequently this population is updated by means of randomized processes of recombination, mutation, and selection, which are inspired by biological evolution. Each individual is evaluated according to a fitness function that depends on the problem to be solved. The selection process is completely deterministic and favors fit individuals from the current population to reproduce in the next generation. For $(\mu+\lambda)$-selection, the μ best individuals from the union of μ parents and λ offspring are selected as parents to form the next generation. For (μ, λ)-selection, the μ best individuals are selected only from the λ offspring; where $\mu < \lambda$.

The recombination process allows combining information from different members of the population to create offspring. Typical mechanisms include discrete recombination, which creates two offspring vectors from two parent vectors copying selected elements from each parent, and intermediate recombination, which commonly uses an arithmetic average, possibly weighted, or extrapolation in the direction of increasing fitness.

Mutation consists of generating random changes in an individual and often provides new relevant information. Mutation is applied independently to each object variable of an individual while strategy parameters may be mutated using a multiplicative logarithmic Gaussian process as shown below:

$$\sigma_i' = \sigma_i \times \exp(N(0,1) + N_i(0,1))$$

$$x_i' = x_i + \sigma_i \times N(0,1)$$

where $N(0,1)$ is a normally distributed random variable having an expectation of zero and a standard deviation of one; $N_i(0,1)$ indicates that the random variable is sampled every time the index i changes.

3 Proposed Algorithm to Find Initial Search Point

In this section we present our novel method to compute a crustal velocity model of the earth using a $(\mu+\lambda)$-ES. The test volume of the Earth's crust is discretized as a very large 3D grid of vertices, where each vertex represents a crustal velocity value. The goal of this method is to find a 1D velocity model, or its equivalent flat-layered 3D model, with minimal rms residuals between calculated first arrival times and traveltimes measured from waves sensed by the receivers. Since the number of vertex velocities to compute in a 3D model is very large, our first goal is to find a 1D model where each layer in the depth dimension has a constant lateral velocity and each layer satisfies the velocity constraint $v_i < v_j$ at depths d_i and d_j respectively, where $d_i < d_j$, for any two layer velocities v_i and v_j in the model. To reduce the load on the ES algorithm further, we search only for the 1D model inflection points and calculate the remaining points using linear interpolation

To find the inflection points of the 1D model, we use a $(\mu + \lambda)$-selection ES because the number of parameters to evolve is small and parameters are real values within the input velocity range. Thus, an individual represents a small set of layer depth values of the 3D model and their associated values of constant lateral velocity.

3.1 Individual Representation

Each individual has two parts with p elements. The first part contains p layer velocity values and the second part contains p layer depths as shown in Figure 1. Each part of an individual is a sorted set of values to meet the velocity constraint expressed above. The number of 1D model inflexion points to search is provided by the user. The values of velocity of the layers between a pair of inflexion points are calculated by linear interpolation.

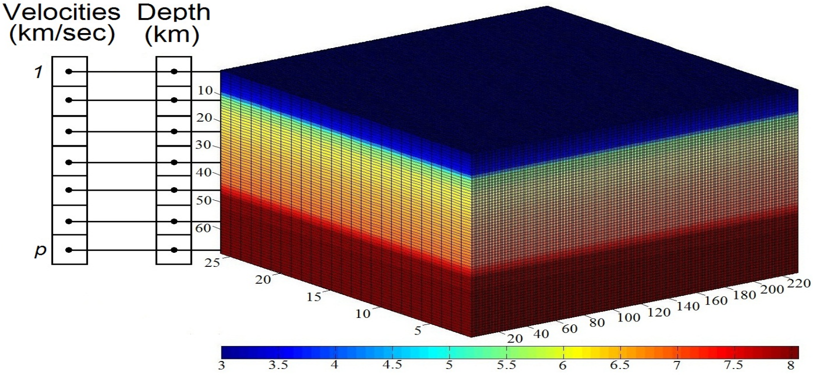


Fig. 1. 1D and 3D representation of an evolution strategy individual showing constant velocity values in each depth layer

3.2 Fitness Function

To evaluate each individual, which represents a 1D velocity model, we compute the first arrival times throughout the discretized velocity model and then we back trace seismic waves from receivers to sources through the resulting discretized traveltime field using the seismic tomography procedures deve—loped by Vidale [16] and Hole [8]. Computed first arrivals of the back-traced waves are compared against the observed first arrival times from the field experiment. The average of the root mean squared differences between observed and calculated times, referred to as the residuals, is calculated for all sensors and shotpoints and returned as a fitness value. The fitness function is computed with the following equation:

$$f(I_p) = \frac{1}{M}\sum_{i=1}^{M}\sqrt{\frac{1}{N}\sum_{j=1}^{N}(t_{oij} - t_{cij})^2}$$

where I_p is an individual with p inflexion points, M is the number of shotpoints, N is the number of sensors, t_{oij} is the measured first arrival traveltime between shotpoint i and sensor j , and t_{cij} is the computed first arrival traveltime counterpart of t_{oij}.

To accelerate the ES search, each individual is sorted before evaluation using the velocity constraint at the beginning of this section. Although the crustal structure may contain soft material that reduces the velocity of seismic waves, this is not considered by the ES to generate the 1D velocity model.

4 Experimental Results

In this section we present the results achieved by applying the presented hybrid approach to the data collected from an experiment performed in the Potrillo volcanic field [1]. With the reference 1D model provided by a geologist, the observed traveltimes, the shotpoints, the receiver locations, and the model geometry, we ran the Vidale-Hole seismic tomography algorithm. The first iteration of the algorithm produced a velocity model with an rms residual time of 0.2476s. The final model obtained had an overall rms residual time of 0.0770s.

To determine whether our hybrid approach can produce a similar or better result without using an initial 1D model provided by a domain expert, we applied an evolution strategy to find an initial 1D model using the velocity constraint and the fitness function described in the previous section.

$(\mu + \lambda)$-ES was applied to find 1D models with 6, 7, 8 and 9 inflexion points and we used interpolation to generate initial 3D velocity models for the seismic tomography procedure used as the second step of the hybrid algorithm. The ES parameters were an initial population of $\mu = 30$ individuals; each individual consisted of $2 * p$ elements, where p is the number of inflexion points. The first p elements were sorted real numbers in the user-input velocity range of 1 to 10 Km/s and the second p numbers were sorted integer numbers in the user-input depth range of 1 to 69 Km; objective variables were randomly generated

Table 1. Experimental results for each 1D model

1D Model Generator	RMS RESIDUAL TIME (seconds)	
	1D Model	Final Model
Expert reference	0.2476	0.0770
ES (6 points)	0.2518	0.0813
ES (7 points)	0.2149	0.0853
ES (8 points)	0.1880	0.0765
ES (9 points)	0.1986	0.0787

with a uniform probability density function. Initial standard deviations used for mutation were also random numbers, but they were generated with a normally distributed probability density function with zero mean and standard deviation equal to 0.5.

Each ES test ran for 50 generations. In each generation, a new population of $\lambda = 30$ offspring was reproduced by means of mutation over 60% of the individuals; the remaining 40% were generated by recombination using the following types of recombination operators: intermediate, discrete, global intermediate and global discrete operations. Ten percent was allocated for individuals of each operator type. Before passing each individual to a new population, individual elements were sorted with respect to both depth and velocity.

Table 1 shows the results of tests using from 6 to 9 inflection points and a test with a 1D model provided by an expert geologist applied to data; the values shown in the table are the average of five runs for each case. The first column identifies the generator of the 1D model used for the seismic tomography algorithm; the second column shows the fitness value or rms residual time average of the 1D models; and the third column shows the values of the final rms residuals obtained using each of the 1D models as input for the tomographic algorithm.

We can observe that the all the 1D models produced by ES are competitive with those produced by the expert, and that the model with 8 inflection points produces the best search point and final rms residual time of the seismic tomography algorithm. 1D models provided by ES after 50 generations have rms residual times below 0.26 s, which is a user-defined threshold. In addition, initial rms residuals of 1D models with 7 or more inflection points are below the value of the model provided by the expert. Final rms residual values are clustered around the value obtained using the reference model, but using 1D models with 8 and 9 inflection points produced the best final residual values.

Figure 2 shows a comparison of the model generated by the expert and the best 1D model generated by ES. The models are remarkably similar and lead to similar initial and final errors, with the ES-generated model having superior performance. Figure 3 shows the performance of the Vidale-Hole seismic tomography algorithm using the starting point provided by the expert and the one obtained by ES. It can be seen that results are similar, with ES producing smaller residuals at every point of the search.

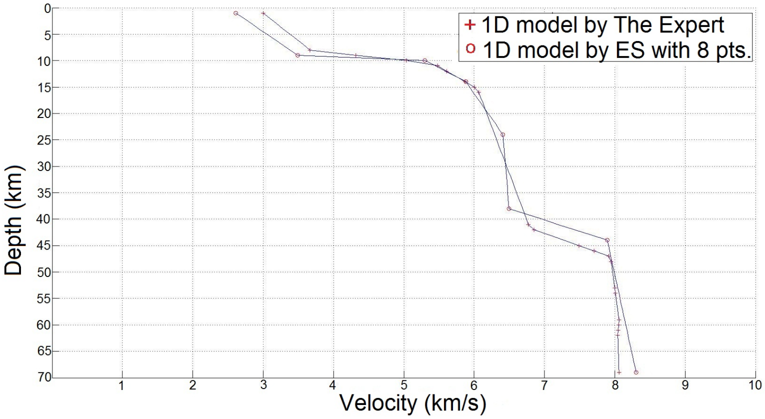


Fig. 2. The expert-generated 1D model and the best model generated by our algorithm, which uses eight inflexion points

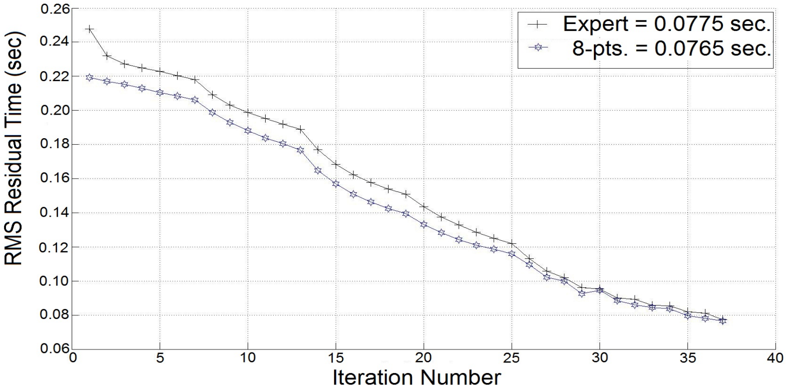


Fig. 3. Residuals at different points of the search generated by the Vidale-Hole algorithm using the expert-provided and ES generated starting points.

5 Conclusion and Future Work

We proposed a two-step hybrid algorithm that combines a $(\mu + \lambda)$-selection evolution strategy and a first arrival traveltime seismic tomography algorithm to generate a 3D crustal velocity model of the Earth. The ES was applied to find a 1D velocity model comprised of a set of layers of constant lateral velocity. The ES was used to search for inflexion points of the 1D velocity model and the full model vertex velocity values were interpolated. Using an ES to generate an initial 1D velocity model eliminates the need for a 1D model provided by an expert geologist for the seismic tomography algorithm and enables users to define a target initial rms residual for the seismic tomography algorithm. The latter point is especially important because large initial rms residuals in the model tend to be associated with lack of model convergence. We tested this method using 6 to 9 inflexion points per 1D model. Our results show that a higher number of inflexion points in the 1D model tends to decrease the rms residual time for the model. However, using more inflexion points translates into a larger search space, which results in longer computation times to find a good solution.

In this work we searched for an initial model for the seismic tomography algorithm with constant lateral velocity layers. Future work includes generating a model with three-dimensional regions of constant lateral velocity using an evolutionary algorithm with recursive subdivision of the search space. Additionally, we will experiment with the analysis of other seismic data sets that are currently being generated by our group and others.

Acknowledgements. This material is based upon work supported in part by the National Science Foundation under CREST Grant No. HRD-0734825 and Grant No. CNS-0923442. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

References

1. Averill, M.G.: A Lithospheric Investigation of the Southern Rio Grande Rift. PhD thesis, Department of Geological Sciences, University of Texas at El Paso, El Paso, TX (2007)
2. Bäck, T., Hoffmeister, F., Schwefel, H.-P.: A survey of evolutionary strategies. In: Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, Inc. (1991)
3. Başokur, A.T., Akça, I., Siyam, N.W.A.: Hybrid genetic algorithms in view of the evolution theories with application for the electrical sounding method. Geophysical Prospecting 55(3), 393–406 (2007)
4. Bezada, M.J., Zelt, C.A.: Gravity inversion using seismically derived crustal density models and genetic algorithms: an application to the Caribbean-South American Plate boundary. Geophysical Journal International 185(2) (2011)
5. Chang, S.-J., Baag, C.-E.: Crustal structure in southern korea from joint analysis of regional broadband waveforms and travel times. Bulletin of the Seismological Society of America 96(3), 856–870 (2006)
6. Fuentes, O., Nelson, R.C.: Learning dextrous manipulation strategies for multi-fingered robot hands using the evolution strategy. Machine Learning 31, 223–237 (1998)
7. Göktürkler, G.: A hybrid approach for tomographic inversion of crosshole seismic first-arrival times. Journal of Geophysics and Engineering 8(99) (2011)
8. Hole, J.A.: Nonlinear high-resolution three-dimensional seismic travel time tomography. Journal of Geophysical Research 97(B5), 6553–6562 (1992)
9. Kishore, N.N., Munshi, P., Ranamale, M.A., Ramakrishna, V.V., Arnold, W.: Tomographic reconstruction of defects in composite plates using genetic algorithms with cluster analysis. Research in Nondestructive Evaluation 22(1), 31–60 (2011)

10. Lanz, E., Maurer, H., Green, A.G.: Refraction tomography over a buried waste disposal site. Geophysics 63(4), 1414–1433 (1998)
11. Liu, L., Gu, T.: Seismic non-destructive testing on a reinforced concrete bridge column using tomographic imaging techniques. Journal of Geophysics and Engineering 2(23) (2005)
12. Mosegaard, K., Sambridge, M.: Monte Carlo analysis of inverse problems. Inverse Problems 18(3) (2002)
13. Polymenakos, L., Papamarinopoulos, S.P.: Exploring a prehistoric site for remains of human structures by three-dimensional seismic tomography. Archaeological Prospection 12(4), 221–233 (2005)
14. Potty, G.R., Miller, J.H.: Tomographic mapping of sediments in shallow water. IEEE Journal of Oceanic Engineering 28(2), 186–191 (2003)
15. Rechenberg, I.: Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Frommann-Holzboog, Stuttgart (1973)
16. Vidale, J.E.: Finite-difference calculation of traveltimes in three dimensions. Geophysics 55(5), 521–526 (1990)

Relationship between Petri Nets and Cellular Automata for the Analysis of Flexible Manufacturing Systems

Irving Barragán, Juan Carlos Seck-Tuoh, and Joselito Medina

Universidad Autónoma del Estado de Hidalgo, Centro de Investigación Avanzada en
Ingeniería Industrial, Carr. Pachuca Tulancingo Km 4.5, Mineral de la Reforma
Hidalgo 42184, México
{irvingb,jseck,jmedina}@uaeh.edu.mx

Abstract. In this paper an association between Petri nets (PN) and cellular automata (CA) is proposed to analyze the global dynamics of flexible manufacturing systems (FMS). This relation is carried out taking into account the discreteness in the dynamics of both PN and CA. In particular, generalized PN as well as one-dimensional CA are used. The work consists in modeling with PN both a single process with a shared resource and two parallel processes with several shared resources. The PN models are simplified by reduction rules and then the corresponding one-dimensional CA is obtained. Finally, the global dynamics of the FMS modeled is described by using the analysis methods of CA.

Keywords: Petri nets, cellular automata, flexible manufacturing systems.

1 Introduction

CA are abstract dynamical systems whose evolution is discrete in space and time. As a modeling tool, CA have been applied in the study of ecological systems, chemical systems, among others [7]. In this regard, one-dimensional CA are the most studied kind of CA and there exists methods contrived to study their dynamics which are mainly based on diagrams. On the other hand, PN are bipartite and directed graphs whose dynamics is discrete and deterministic in the generalized case, [18]. Due to the graphical and mathematical features, PN are a powerful modeling and analytical tool for systems which are concurrent and asynchronous such as FMS, [2], [22], [24] where PN are useful both to describe the dynamics of such systems, [13], and to prevent undesirable situations such as deadlocks, [3]. Nevertheless, some problems arise in the analysis of FMS when using the classical PN analysis methods. On the one hand, such methods can be only applied to specific PN and, on the other hand, some of the analysis methods, like the coverability tree, are not able to manage large PN models due to the state-space explosion.

The association between PN and CA has been explored in [4], [5] for modeling an ecological system. In [11] a self-timed CA is modeled with high-level PN and

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 338–349, 2013.

in [14] self-similar PN and CA are used to simulate Turing machines. In addition, in [16], [17] is proposed the use of CA to model FMS whose dynamics is firstly described by a net which has a similar dynamics to PN. In all these works, PN derivations are used to model the corresponding systems and in some cases the classical dynamics of CA is modified. The relationship between CA and PN we propose in this paper is carried out by using generalized PN and one-dimensional CA in such a way that their usual dynamics is maintained. Then, we describe the global dynamics of FMS by means of the CA analysis methods which will allow a quick and simplified way to determine some dynamical features of the FMS such as reversibility and liveness without dealing with the state-space explosion.

Additionally, we consider two kinds of FMS models to exemplify the process by which the one-dimensional CA are obtained. These FMS models are based on one of the main characteristics of such systems: shared resources. The first example is about a single process which consists of a manufacturing cell with one shared resource. The second example deals with a pair of parallel processes with several shared resources. A PN model is firstly designed for the single process as well as for the pair of parallel processes by using the most common structures to model manufacturing systems. Then, the PN models are simplified by means of PN reduction rules in order to obtain homogeneous subnets in such a way that each subnet can have the same number of markings. Such subnets will represent the cells of the CA. Furthermore, the places of the reduced PN models are considered to have a fixed capacity in the number of tokens they can hold in order to facilitate the definition of the number of states of the CA. Finally, the CA evolution rule is determined according to the dynamics of the PN.

The paper is organized as follows: In Sect. 2 and Sect. 2.2 we give the basic concepts of CA and PN. In Sect. 3 it is explained the process by which the one-dimensional CA are obtained from the PN models of the FMS examples. In Sect. 4 the CA analysis methods are applied to describe the global dynamics of the FMS modeled. Finally, in Sect. 5 conclusions of the work are given.

2 Basic Concepts of Cellular Automata and Petri Nets

2.1 Cellular Automata

CA consist of a configuration of cells C, also known as the global state, where each cell can take a state from a finite set of states $\mathcal{S}$, such that $|\mathcal{S}| = k$. Particularly, in a one-dimensional CA any configuration is a linear array of cells, that is to say $C = c_1c_2\dots c_i\dots$, where each cell forms a neighborhood $\eta(c_i)$ of size $2r+1$ such that r is the neighborhood radius. Usually, a one-dimensional CA can be described by the well-known Wolfram's notation (k, r), [20]. The dynamics of a one-dimensional CA is determined by the evolution rule $\varphi : \mathcal{S}^{2r+1} \to \mathcal{S}$ which assigns a new state to each cell c_i^{t} of C^{t} which is the global state at time t. In this way, $c_i^{\mathrm{t}+1} = \varphi(\eta(c_i^{\mathrm{t}}))$. In spite of C can have an infinite number of cells, from a practical point of view the array is taken as finite, that is $C = c_1c_2\dots c_l$ where l is the length of the configuration. Consequently, periodic boundary conditions are taken for the cells located at the ends of the array such that C is closed forming a ring in such a way that each cell form a neighborhood of the same size. Furthermore, since φ transforms a configuration C into C', then $\Phi : \mathcal{C} \to \mathcal{C}$ defines a global evolution rule induced by φ where $\mathcal{C}$ is the set of all possible configurations. In this regard, let $\mathcal{A} = \{C \mid \Phi^{-1}(C') = C\}$ be the set of ancestors of C' such that $C', C \in \mathcal{C}$, then the CA is reversible if $|\mathcal{A}| = 1$, [6], [15]. Nevertheless, if $|\mathcal{A}| > 1$, then $G \subset \mathcal{C}$ is a set of configurations which have no ancestors and is called *the Garden of Eden*.

There exists some classical methods that can be used to analyze the dynamics of one-dimensional CA like the subset and pair diagrams which are used in this paper. The subset diagram is derived from the de Bruijn diagram which can be defined by $\mathsf{B}(\mathcal{B}, \mathcal{E})$ where $\mathcal{B} = \{\mathrm{u} \mid \mathrm{u} \in \mathcal{S}^{2r}\}$ is a set of nodes and $\mathcal{E} = \{(\mathrm{u}, \mathrm{v}) \mid \mathrm{u}, \mathrm{v} \in \mathcal{B}\}$ is a set of links, [19]. In this way, let $\mathrm{u} = ax$ and $\mathrm{v} = xb$ be two nodes in the de Bruijn diagram such that $x \in \mathcal{S}^{2r-1}$ and $a, b \in \mathcal{S}$, then u and v can be linked if there exists $\mathrm{u} \odot \mathrm{v} = axb$, where $\odot$ defines an overlapping operation between two nodes and $axb \in \mathcal{S}^{2r+1}$. Thus, each link is labeled with $c \in \mathcal{S}$ such that $\varphi(axb) = c$. Besides, an evolution matrix can be obtained from the de Bruijn diagram in such a way that the nodes constitutes the rows and the columns of the matrix and each entry is either determined by $\varphi(\mathrm{u} \odot \mathrm{v})$ if the link exists or by a dot if the link does not exist, [9].

Therefore, the subset diagram is composed of the subsets of $\mathcal{B}$. In this way, let $W, Z \subseteq \mathcal{B}$ be two nodes of the subset diagram, then W and Z are linked if for each node $\mathrm{w} \in W$ there exists another node $\mathrm{z} \in Z$ such that $\varphi(\mathrm{w} \odot \mathrm{z}) = c$ and c labels the link. Finally, the cycle diagram consists in graphing the evolution trajectory of all the k^l possible configurations $C \in \mathcal{C}$. Hence, a given configuration C can appear again in the trajectory of its evolution forming a cycle, [21]. Nevertheless, if there are configurations that do not appear in the cycle then they belong to a sequence of configurations that leads to the cycle. Such a sequence forms a branch of the cycle and, therefore it has an end which is a configuration which has no ancestors.

2.2 Petri Nets

A PN consists of the sets of nodes $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$, called places and transitions, respectively [1], [12], such that $m, n \in \mathbb{Z}^+$ where $\mathbb{Z}^+ = \{1, 2, \ldots\}$. Moreover, $\mathcal{P} \cup \mathcal{T} \neq \emptyset$ and $\mathcal{P} \cap \mathcal{T} = \emptyset$. The nodes are connected by a set of directed arcs defined by $\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$. In addition to this, $W : \mathcal{F} \to \mathbb{N}$ is a weight function and $M_0 : \mathcal{P} \to \mathbb{N}$ defines the initial marking, where $\mathbb{N} = \{0, 1, 2, \ldots\}$. In this regard, $\mathbf{N} = (\mathcal{P}, \mathcal{T}, \mathcal{F}, W)$ is the structure of a PN whereas $(\mathbf{N}, M_0)$ is the PN with a given initial marking, [10]. Furthermore, any marking M is represented by an $m \times 1$ vector such that $\boldsymbol{M}(p_i) = s$ is the number of tokens of place p_i where $i = 1, \ldots, m$ and $s \in \mathbb{N}$. In this way, it is possible to associate a maximum number of tokens to each place of a PN such that, for each place $p \in \mathcal{P}$, $K : \mathcal{P} \to \mathbb{Z}^+$ is the function that assigns the maximum number of tokens that a place can hold at any marking.

As for the PN dynamics, let ${}^\bullet t = \{p \mid W(p, t) > 0\}$ and $t^\bullet = \{p \mid W(t, p) > 0\}$ be the sets of input and output places of t, respectively. Then for a given net

$(\mathbf{N}, M_0)$ its dynamics or marking change is carried out by applying the *enabling and firing rule.* A transition t is *enabled* at the marking M_d $(d \in \mathbb{N})$ if $\forall p \in {}^{\bullet}t$, $M_d(p) \geq W(p,t)$ and, in the case of PN of finite capacity, $\forall p \in t^{\bullet}$, $M_d(p) \leq K(p) - W(t,p)$. As a result, if t is enabled at the marking M_d then t can be *fired* transforming M_d into M_{d+1} such that $M_{d+1}(p) = M_d(p) - W(p,t) + W(t,p)$.

PN properties are classified into dynamical and structural ones. Dynamical properties are those which depends on the initial marking whereas structural properties are independent of the initial marking. In this paper we are interested in some dynamical properties which are reachability, reversibility and liveness. In this regard, let $\mathcal{M}(M_0)$ be the set of reachable markings of a PN from M_0 and $\mathcal{L}(M_0)$ be the set of firing sequences executable from M_0. Then, a marking M_q is *reachable* from M_0 if there exists a firing sequence $\sigma = t_1 t_2 \cdots t_q$ that leads to M_q. This is denoted by $M_0 \overset{\sigma}{\rightarrow} M_q$, such that $\sigma \in \mathcal{L}(M_0)$. In this way, a PN is *reversible* if for each marking $M \in \mathcal{M}(M_0)$, M_0 is reachable from M. A PN is *live* if $\forall t \in \mathcal{T}$ and for any marking $M \in \mathcal{M}(M_0)$, t appears in some sequence $\sigma \in \mathcal{L}(M_0)$ such that $M_0 \overset{\sigma}{\rightarrow} M$, [18].

Among the basic methods that can be used to analyze a PN, reduction (or augmentation) rules are procedures to simplify PN with a complex structure but keeping some dynamical properties like liveness. The following reduction/augmentation rules are the basic ones, [23]:

Rule 1. Reduction/augmentation of series places.
Rule 2. Reduction/augmentation of series transitions.
Rule 3. Reduction/augmentation of parallel places.
Rule 4. Reduction/augmentation of parallel transitions.
Rule 5. Elimination/addition of self-loop places.
Rule 6. Elimination/addition of self-loop transitions.

Dynamical properties such as reachability, liveness as well as reversibility are of special interest for modeling and analyzing the FMS performance. For instance, liveness is a sign of absence of deadlocks which arise from the competition for the access to shared resources. Reachability can be used during the design stage to identify whether the modeled system can reach a particular state and reversibility is connected with error recovery.

3 Construction of One-Dimensional CA from PN Models

In this section we explain how to obtain a one-dimensional CA from PN models of FMS. In order to carry out this task, two examples of FMS are considered. These examples comprise the most common structures and features of FMS. The first example is a manufacturing cell, shown in Fig. 1 which is composed of a robot which is shared by three machines named M1, M2 and M3. The second example consists of two parallel processes where two types of parts can be machined, Parts A and Parts B. In order to perform the processing of both kinds of parts, machines M1, M2, M3, M4 and M5 must be shared as is illustrated in Fig. 2.

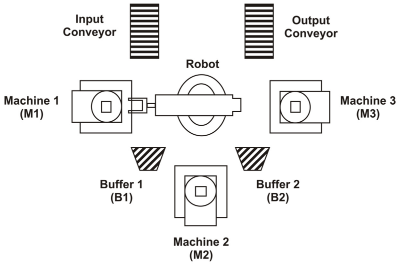


Fig. 1. A manufacturing cell with one shared resource

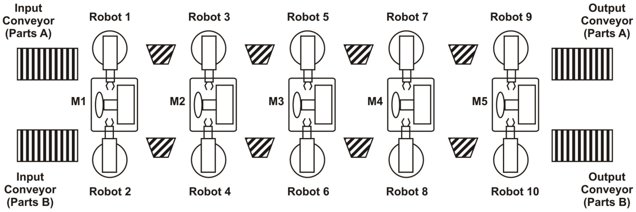


Fig. 2. Parallel processes with five shared resources

The PN models of the corresponding examples of FMS and their initial markings are depicted in Fig. 3 and Fig. 4, correspondingly. We consider that only one part can be available at the beginning of each process in such a way that any other can arrive until the first one leaves the system.

In order to obtain a one-dimensional CA from each PN model, the reduction rules 2 and 5 shown in Figs. 5a and 5b, respectively, are applied to such PN models. The reason to simplify the PN models is that it is desired to obtain homogeneous subnets to form the cells of the automaton and, according to the number of possible markings of each subnet, the states of the CA can be specified. In this regard, to facilitate the definition of the number of states of the CA, the places of the reduced PN models are limited to hold one token such that for any place p, $K(p) = 1$. Furthermore, the neighborhood radius for each CA obtained from the reduced PN is taken as $r = 1$.

3.1 Example 1: Obtaining a One-Dimensional CA for a Single Process in Series

A first reduction for the PN shown in Fig. 3 is carried out by applying the Rule 2, and the result is shown in Fig. 6a. Then, the place p_{16} is eliminated by Rule 5, obtaining the net shown in Fig. 6b. From this net it can be obtained two kinds of subnets. One subnet has one place and the other one has two places. In this way, these subnets constitutes the cells of the CA as it is shown in Fig. 7, whereas the possible markings in each subnet determine the number of states of the CA.

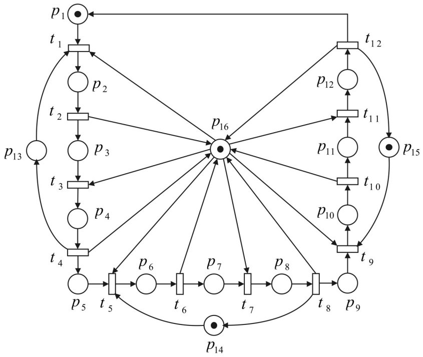


Fig. 3. PN model of the manufacturing cell with a shared resource

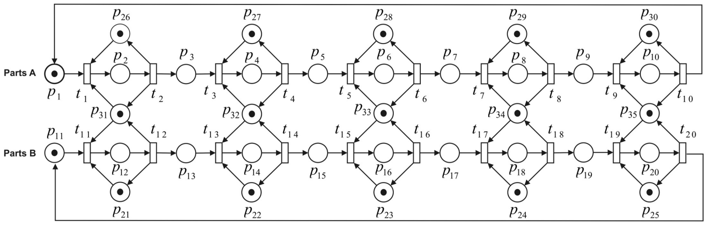
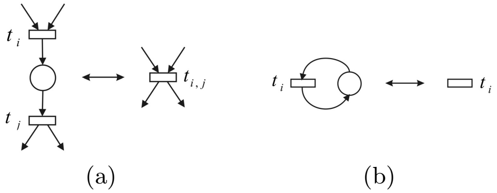
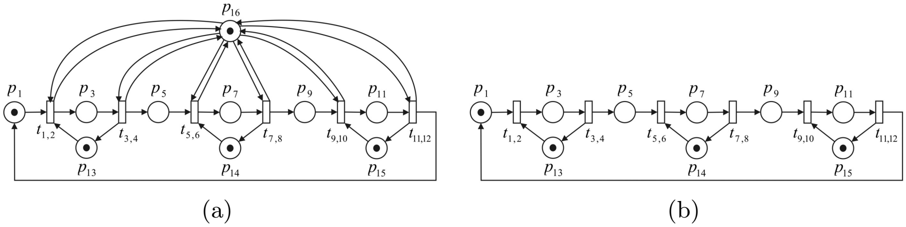


Fig. 4. PN model of the parallel processes with five shared resources



(a) (b)

Fig. 5. (a) Rule 2: Reduction/augmentation of series transitions; (b) Rule 5: Elimination/addition of self-loop places



(a) (b)

Fig. 6. Reductions of the PN of Fig. 3 by applying (a) Rule 2 and (b) Rule 5

The CA states are taken as 0 and 1, such that $\mathcal{S} = \{0, 1\}$. Hence, for subnets with one place the state is 0 whenever the place is not marked, which means that there is no available parts, and 1 if the place is marked. For subnets with two places take as example the one which has the places p_{13} and p_3. Thus, when p_3 is marked it means that the corresponding operation is being performed and thus the state is taken as 1. In contrast to this, when place p_{13} is marked the state is 0 which means that the operation is not being executed. The evolution rule φ is determined according to the dynamics of the PN of Fig. 7. In other words, consider $\eta(c_2) = c_1c_2c_3$, since p_3 and p_{13} are related to p_1 and p_5 by transitions $t_{1,2}$ and $t_{3,4}$ respectively, then if either $t_{1,2}$ or $t_{3,4}$ are fired they will cause a marking change of p_3 and p_{13} and, as a result, c_2 change its state. In this way, φ is described by the evolution matrix defined in Equation (1). Thus, the PN of Fig. 7 is simulated by a CA $(2, 1)$ where the initial configuration is $C^0 = 100000$ and the corresponding marking is $M = (1\,0\,1\,0\,0\,1\,0\,0\,1)^{\mathrm{T}}$.

$$\begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \left[\begin{array}{cccc} 0 & 0 & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & 0 & 1 \end{array}\right] \end{array} \tag{1}$$

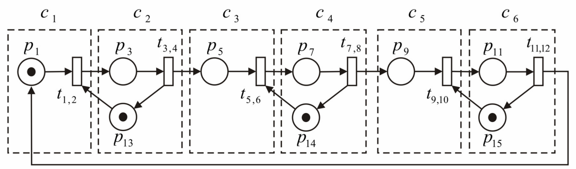
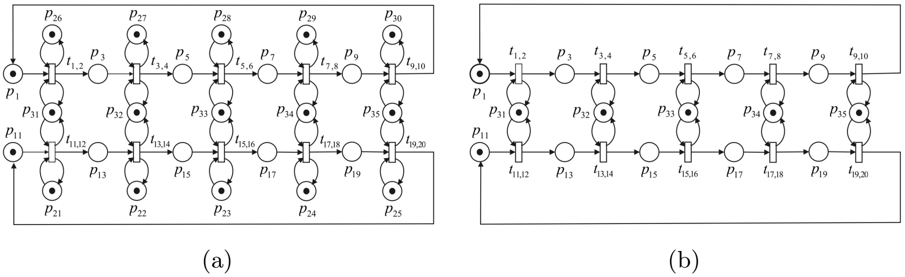


Fig. 7. Cells for the one-dimensional CA which simulates the PN of Fig. 6b

3.2 Example 2: Obtaining a One-Dimensional CA for Processes in Parallel.

In a similar way to Example 1, the PN model of Fig. 4 is reduced by firstly applying the Rule 2 and then the Rule 5 such that the result is shown in Fig. 8. The subnets that form the cells of the CA are depicted in Fig. 9 from which it is observed that there can be four possible markings for each subnet.

Thus, the set of states of the CA is defined by $\mathcal{S} = \{0, 1, 2, 3\}$ and the correspondence between the markings and the states is described in Table 1, taking as example the subnet with the places p_1, p_{11}, p_{31} which represent the cell c_1. With respect to the PN dynamics, it is established that whenever occurs a conflict for the access to a shared resource, Parts A must be processed first. Moreover, the enabled transitions are fired at the same time if the places do not exceed the capacity of one token and if such transitions are not in conflict. Considering this, the evolution rule is defined according to the marking change of the PN and is



(a) (b)

Fig. 8. Reductions for the PN model of Fig. 4 by applying (a) Rule 2 and (b) Rule 5

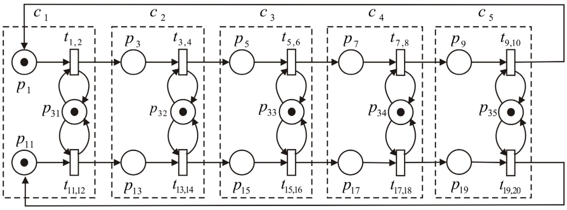


Fig. 9. Cells for the CA $(4, 1)$ that represents the dynamics of the processes in parallel

defined by the evolution matrix shown in Equation (2). Hence, a CA $(4, 1)$ is used to simulate the PN of Fig. 9 where the initial configuration is $C^0 = 30000$.

$$
\begin{array}{c|cccccccccccccccc|}
 & 00 & 01 & 02 & 03 & 10 & 11 & 12 & 13 & 20 & 21 & 22 & 23 & 30 & 31 & 32 & 33 \\
00 & 0 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
01 & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
02 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 2 & 2 & \cdot & \cdot & \cdot & \cdot \\
03 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2 & 1 & 2 & 3 \\
10 & 1 & 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
11 & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
12 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & 3 & 3 & \cdot & \cdot & \cdot & \cdot \\
13 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2 & 1 & 2 & 3 \\
20 & 2 & 2 & 2 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
21 & \cdot & \cdot & \cdot & \cdot & 2 & 3 & 2 & 3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
22 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 2 & 2 & \cdot & \cdot & \cdot & \cdot \\
23 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2 & 1 & 2 & 3 \\
30 & 2 & 2 & 2 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
31 & \cdot & \cdot & \cdot & \cdot & 2 & 3 & 2 & 3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
32 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & 3 & 3 & \cdot & \cdot & \cdot & \cdot \\
33 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2 & 1 & 2 & 3
\end{array}
\tag{2}
$$

Table 1. States that can take each cell according to the markings of each subnet of Fig. 9, taking the possible markings of p_1, p_{31} and p_{11} as examples.

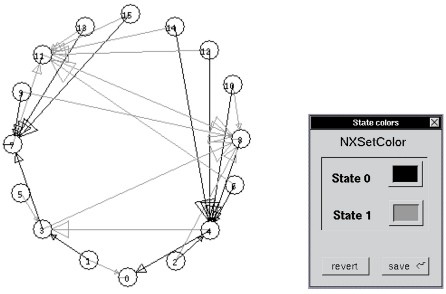
$M(p_1)$	$M(p_{31})$	$M(p_{11})$	State
0	1	0	0
0	1	1	1
1	1	0	2
1	1	1	3

4 Applying the CA Analysis Methods to Describe the FMS Dynamics

The analysis of the global dynamics of the PN models of Figs. 7 and 9 is carried out by means of the CA analysis methods explained in Sect. 2.2. These diagrams were obtained with the help of the NXLCAU software developed by H. V. McIntosh, [8].

4.1 Analysis of the FMS of Example 1

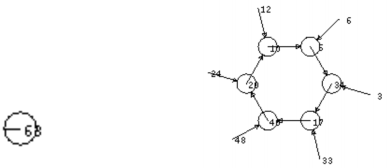
In Fig. 10a is shown the subset diagram of the CA $(2,1)$ that simulates the PN of Fig. 7. From this diagram it is possible to identify those configurations that can be formed or not in the CA. For instance, if there is a path from node 15 to node 0, that is to say, from the full set to the empty set, then the formed configuration does not have any ancestors and it only can appear as an initial configuration.



(a) (b)

Fig. 10. (a) Subset diagram for the CA (2,1) of Example 1. (b) State color settings of NXLCAU for the subset diagram.

In this sense, the path $15 \xrightarrow{1} 11 \xrightarrow{1} 11 \xrightarrow{0} 8 \xrightarrow{0} 4 \xrightarrow{0} 0 \xrightarrow{1} 0$ forms a configuration that represent the marking $M = (1\,1\,0\,0\,0\,1\,0\,1\,0)^{\mathrm{T}}$. Therefore, the formed configuration belongs to the Garden of Eden and in the PN the corresponding marking cannot be formed because of the places of the PN can have at most one token and all the enabled transitions are fired simultaneously, if they are not in conflict. Even though there are configurations that can be formed in the CA according to the subset diagram, these configurations represent markings that are undesirable in the PN. For example, the marking $M = (1\,1\,1\,1\,1\,1\,1\,1\,1)^{\mathrm{T}}$ is a deadlock since any transition can be fired, considering that the PN is of finite capacity. In the CA evolution the corresponding configuration leads to a cycle of only one node as it is shown in Fig. 11a.



(a) (b)

Fig. 11. Cycle diagrams of the CA (2,1) with $l = 6$

Cycle diagrams are useful for determining which configurations are part of a cycle or even which configurations lead to a cycle and in how many steps. For example, Fig. 11b shows a cycle diagram where nodes 10, 5, 34, 17, 40, 20, 10 form the cycle and nodes 12, 6, 3, 33, 48 and 24 lead to the cycle in one step. That is to say, the configuration C = 000110 (node 6), which represents the marking $M = (0\,0\,1\,0\,1\,0\,1\,0\,1)^{\mathrm{T}}$, evolves into the configuration C = 000101 (node 5) in only one step so that the marking $M = (0\,0\,1\,0\,1\,0\,0\,1\,0)^{\mathrm{T}}$ is reachable immediately after $t_{9,10}$ is fired at $M = (0\,0\,1\,0\,1\,0\,1\,0\,1)^{\mathrm{T}}$ in the PN of Fig. 7.

4.2 Analysis of the FMS of Example 2

Cycle diagrams are used here in order to perform the analysis of the PN since it is not possible to show the subset diagram because of its size. In this way, Fig. 12 shows all the cycle diagrams that are obtained for the CA (4,1). It can be observed from Fig. 12 that the first four cycle diagrams (from the left upper corner of the figure to the right) are composed of only one node, which means that the corresponding markings in the PN are deadlocks since the evolution of the CA as well as the marking change in the PN remain unalterable. Similarly,

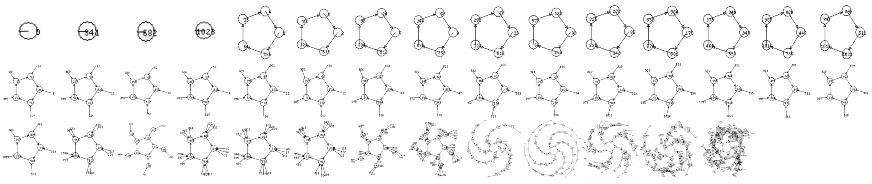


Fig. 12. Cycle diagrams for the CA (4,1) of Example 2 with $l = 5$

for the next eleven cycle diagrams, which are formed by a cycle of five nodes, each of the configurations of the cycles can be initial markings in the PN such that for these markings the PN is reversible.

5 Conclusions

In this paper a relationship between PN and CA was presented in order to carry out the analysis of the global dynamics of FMS. This association was carried out by using generalized PN and one-dimensional CA. Two examples of FMS were considered to illustrate the process by which CA analysis methods are applied to the study of the dynamics of such FMS. This examples consisted in a single process as well as in a pair of parallel processes both with shared resources which is a common feature of FMS. PN models were designed for such FMS examples and then such PN models were simplified by means of PN reduction rules in order to obtain the corresponding one-dimensional CA. Then, by using CA analysis methods, mainly subset and cycle diagrams, the PN models were analyzed to determine which configurations can be formed in the CA and therefore which markings can be reached in the PN under the dynamical conditions imposed to them. In this way, it could be determined, according to the characteristics of the CA dynamics, some features of the FMS like the absence of deadlocks and reversibility. On the whole, CA provided a simplified analysis of the global dynamics of FMS by obtaining one-dimensional CA from the PN models. In spite of that, not all configurations in the evolution of CA are significant markings in the PN dynamics, in the sense that they could not have a practical meaning in the system modeled. Nevertheless, an advantage of the use of CA analysis methods is that, no matter what initial configuration is taken in the PN, it can be determined which will be the dynamics in a global way. Future studies about the analysis of PN dynamics by means of the relationship between PN and CA proposed here will be carried out in depth in a next paper.

References

1. David, R., Alla, H.: Petri Nets for Modeling of Dynamic Systems–A survey. Automatica 30, 175–202 (1994)
2. D'Souza, K.A., Khator, S.: A Survey of Petri Net Applications in Modeling Controls for Automated Manufacturing Systems. Computers in Industry 24, 5–16 (1994)
3. Ezpeleta, J., Colom, J., Martínez, J.: A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems. IEEE Transactions on Robotics and Automation 11, 173–184 (2002)
4. Gronewold, A., Sonnenschein, M.: Asynchronous Layered Cellular Automata for the Structured Modelling of Ecological Systems. In: Hahn, W., Lehmann, A. (eds.) 9th European Simulation Symposium (ESS 1997), pp. 286–290 (1997)
5. Gronewold, A., Sonnenschein, M.: Event-Based Modelling of Ecological Systems with Asynchronous Cellular Automata. Ecological Modelling 108, 37–52 (1998)
6. Kari, J.: Representation of Reversible Cellular Automata with Block Permutations. Mathematical Systems Theory 29, 47–61 (1996)

7. Kari, J.: Theory of Cellular Automata: A Survey. Theoretical Computer Science 334, 3–33 (2005)
8. McIntosh, H.V.: NXLCAU software, `http://delta.cs.cinvestav.mx/˜mcintosh/cellularautomata/Welcome.html`
9. McIntosh, H.V.: Linear Cellular Automata Via de Bruijn Diagrams, `http://delta.cs.cinvestav.mx/~mcintosh/comun/cf/debruijn.pdf`
10. Murata, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of IEEE 77, 541–580 (1989)
11. Penttinen, O.: Modelling and Analysis of Self-timed Cellular Automata Using High-Level Petri Nets. In: Proc. of Workshop on Token based computing (ToBaCo), Satellite Event of the 25-th International Conference on Application and Theory of Petri Nets (2004)
12. Peterson, J.L.: Petri Nets. Computing Surveys 9, 223–252 (1977)
13. Recalde, L., Silva, M., Ezpeleta, J., Teruel, E.: Petri Nets and Manufacturing Systems: An Examples-Driven Tour. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) Lectures on Concurrency and Petri Nets. LNCS, vol. 3098, pp. 742–788. Springer, Heidelberg (2004)
14. Schaller, M., Svozil, K.: Scale-Invariant Cellular Automata and Self-similar Petri Nets. The European Physical Journal B 69, 297–311 (2009)
15. Seck-Tuoh, J.C.: Matrix Methods and Local Properties of Reversible One-Dimensional Cellular Automata. J. Phys. A: Math. Gen. 35, 5563–5573 (2002)
16. Shen, H., Chau, H., Wong, K.: An Extended Cellular Automaton Model for Flexible Manufacturing Systems. The International Journal of Advanced Manufacturing Technology 11, 258–266 (1996)
17. Shen, H., Yan, W.: Modeling Autonomous Assembly Systems and FMS Using Cellular Automata. The International Journal of Advanced Manufacturing Technology 7, 333–338 (1992)
18. Silva, M.: Las Redes de Petri: En la Automática y la Informática. AC, Madrid (2002)
19. Sutner, K.: De Bruijn Graphs and Cellular Automata. Complex Systems 5, 19–30 (1991)
20. Wolfram, S.: Statistical Mechanics of Cellular Automata. Reviews of Modern Physics 55, 601–644 (1983)
21. Wuensche, A., Lesser, M.: The Global Dynamics of Cellular Automata. Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley Publishing Company, USA (1992)
22. Zhang, W., Li, Q., Zha, X.F.: A Generic Petri Net Model for Flexible Manufacturing Systems and Its Use for FMS Control Software Testing. Int. J. Prod. Res. 38, 1109–1131 (2000)
23. Zhou, M.-C., Venkatesh, K.: Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific, Singapur (1999)
24. Zurawski, R., Zhou, M.C.: Petri Nets and Industrial Applications: A Tutorial. IEEE Transactions on Industrial Electronics 41, 567–583 (1994)

An Integrated Strategy for Analyzing Flow Conductivity of Fractures in a Naturally Fractured Reservoir Using a Complex Network Metric

Elizabeth Santiago, Manuel Romero-Salcedo, Jorge X. Velasco-Hernández,
Luis G. Velasquillo, and J. Alejandro Hernández

Instituto Mexicano del Petróleo, Av. Eje Central Lázaro Cárdenas Norte, 152
Col. San Bartolo Atepehuacan, Mexico-City, Mexico CP 07730
{esangel,mromeros,velascoj,lgvelas}@imp.mx,
jahernandezv87@gmail.com

Abstract. In this paper a new strategy for analyzing the capability of flow conductivity of hydrocarbon in fractures associated to a reservoir under study is presented. This strategy is described as an integrated methodology which involves as input data the intersection points of fractures that are extracted from hand-sample fracture images obtained from cores in a Naturally Fractured Reservoir. This methodology consists of two main stages. The first stage carries out the analysis and image processing, whose goal is the extraction of the topological structure from the system. The second stage is focused on finding the node or vertex, which represents the most important node of the graph applying an improved betweenness centrality measure. Once the representative node is obtained, the intensity of intersection points of the fractures is quantified. In this stage a sand box technique based on different radius for obtaining an intensity pattern in the reservoir is used. The results obtained from the integrated strategy allow us to deduce in the characterization of reservoir, by knowing the possible flow conductivity in the topology of the fractures viewed as complex network. Moreover our results may be also of interest in the formulation of models in the whole characterization of the reservoir.

Keywords: flow conductivity of fractures, complex network metric, naturally fractured reservoirs.

1 Introduction

One of the principal challenges in characterization of naturally fractured reservoirs (NFR) in the hydrocarbon industry is the generation of a representative model of the reservoir [1, 2, 3, 4]. This characterization requires putting together different data sources about the whole reservoir [5, 6, 7]. One of the most important problems is the determination of the nature, and disposition of heterogeneities that inevitably occurs in petroliferous formations in order to determine the capability for fluid transport. Different strategies have been developed to tackle this problem. Some authors have focused on the analysis of the properties of the fluid flow [8, 9], others in the

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 350–361, 2013.

modeling and simulation of fracture networks [10], and some others in the analysis of topological properties by applying statistical techniques to the structure where the fluid transport may occur [11]. In this last approach most authors use synthetically generated fracture networks [26]. This paper deals with this last approach, but through the extraction of parameters from original hand-sample images supplied by geologists. These images correspond to a Gulf of Mexico oil reservoir, and will be used as test examples for the determination of network topologies [12, 13]. In particular, we present and discuss the application of a network metric, which computes the number of the shortest paths that pass through a certain node [13]. In this work, a new strategy for the identification of patterns on a set of fracture images is presented. The processing and analysis of fracture hand-sample images uses the methodology KDD (knowledge discovery in database) [14, 27] in order to find patterns embedded in their topologies. We are particularly interested in the importance of a node as related to its topological function within the network. The methodology includes a metric designed to characterize and identify "an important node", and with it a new strategy is implemented to qualitatively asses flow capability from the fracture images. The metric used here applies the notion of the vertex's importance in a graph that depends on many factors to be described later in this work. Finally, a step of evaluation of this node is carried out to estimate the intensity of intersection points of fractures.

The paper is organized as follows. In section 2, the general scheme of the proposed methodology is described. The image preprocessing is also treated and the applied methods are explained. In section 3, the analysis of nodes is evaluated and the description of the improved betweenness centrality metric used is presented. In section 4, we show our results, first applying the methodology to a set of four fracture hand-sample images, and then applying it to a set of 100 fracture hand-sample images. Finally, in section 5, the conclusions and future work are presented.

2 New Methodology for Characterizing the Topology of Fractures in NFR

In the next paragraphs, a novel methodology for the analysis of the topological structure of fracture networks is applied. Our input data are fracture images. The preprocessing and analysis of fractures is applied to a set of 100 hand-sample fracture images of cores (samples of rocks recovered from a formation of interest commonly used for its study and evaluation) of the Gulf of Mexico. These images have a bmp format. Some examples (labeled as Frac-6 and Frac-93) are shown in Fig. 1. Most of the software available for analyzing the topology of complex networks (such as Gephy and Cytoscape) require as input data files that store the adjacency matrix of the graph. Our methodology generate the adjacency matrix through several steps to recover the fracture structure from the original fracture images, and then the application of the necessary operations to characterize intensity patterns of non-terminal nodes (or intersection points) in fractures. We define an intensity pattern as a tendency in the increase of number of nodes in different sizes of circles. The general procedure is described below.

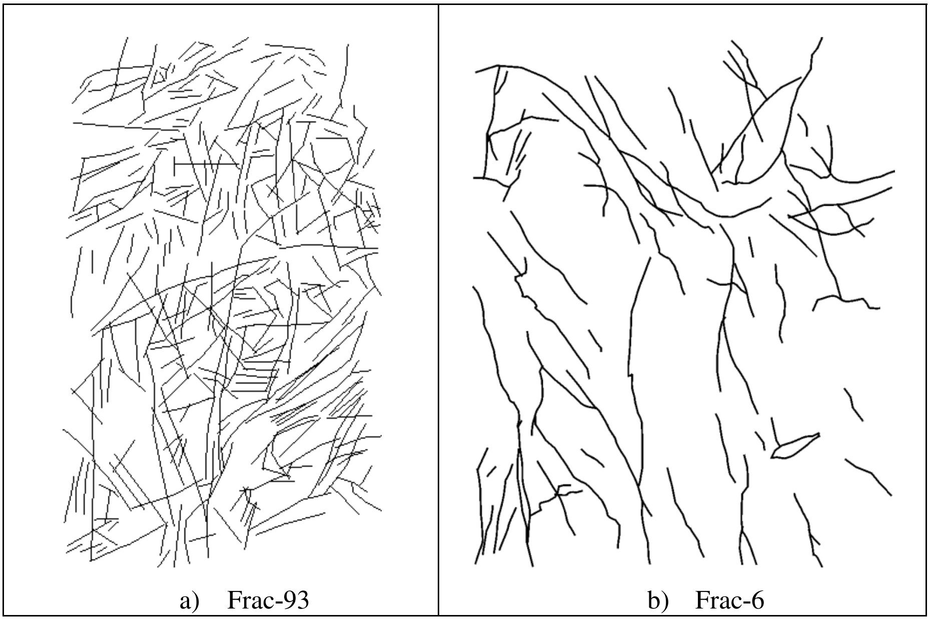
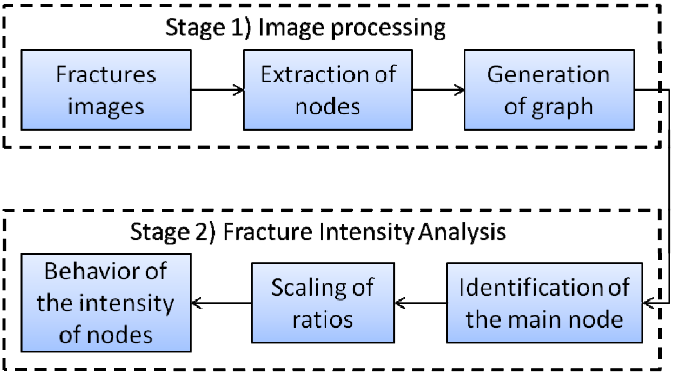


Fig. 1. Hand sample examples of fractures associated to NFR: a) Frac-93, and b) Fract-6 from Jujo-Tecominoacán reservoir

Considering the aforementioned observations, our strategy for identifying patterns in the distribution of the fracture intersections is presented in two stages: 1) creation of a graph from the processed image, and 2) analysis of the intensity of nodes in the fractures. Both are outlined in Fig. 2. The description of the first stage is explained below, and the second stage is detailed in section 3.


Stage 1) Image processing

Fractures images → Extraction of nodes → Generation of graph

Stage 2) Fracture Intensity Analysis

Identification of the main node → Scaling of ratios → Behavior of the intensity of nodes

Fig. 2. General schema of the proposed methodology

The goal of the first stage is to obtain the adjacency matrix representative of the topological structure of the fracture network. We do it in three steps. In the first step the image is processed, applying a skeletonization process for recovering the structure of the fracture in the whole hand-sample image. The second step identifies and obtains the intersections and endpoints of each fracture image, what are known in graph theory as non-terminal and terminal (leaves) nodes, respectively. The third step generates a file that contains the adjacency matrix, where the recovery of all the nodes and edges is done. It requires touring all the paths in the skelotonized fracture in order to find the neighborhood nodes, i. e., to identify which nodes are linked to each node for establishing the connection. Then, this matrix is used for building the graph. The second stage indentifies patterns in the distribution of intersection points in the fractures by means of the application of a centrality metric widely used in complex networks. In the next section, the main idea and the definition of this metric are explained.

3 Fracture Intensity Analysis

The second stage is shown in the dotted lower rectangle in Fig. 2. This stage is subdivided into three steps. An important property of the fracture network is the capability of transport fluids. In this case, the non-terminal nodes or intersections of traces are considered. Thus, the strategy is to analyze different sizes of circular regions from a particular node for evaluating precisely the amount of these nodes in each region. It will allow us to identify any pattern in the increment of different sizes of radius considering the number of intersection points computed. The first step of the second stage is to select the most important node from the set of nodes stored in the adjacency matrix. The earliest intuitive conception of point centrality was based upon the structural properties of centrality. This idea was introduced by Bavelas [15], and an essential tool for the analysis of networks is the centrality index which is based on counting paths going through a node [16]. These measures define centrality in terms of the degree to which a point falls on the shortest path between others, and therefore it has a potential for controlling the communication of the network. One of these metrics is the betweenness centrality measure which was proposed by Freeman [17], and Antonisse [18]. In this work, an improved version [19] of this measure is applied, which includes a more efficient and faster algorithm for large and very sparse networks. This algorithm is based on an accumulation technique that solves the single-source shortest-path problem, and thus exploits efficiently the sparsity of the network incidence matrix. The traditional formulation for computing the betweenness centrality index is accomplished in two steps: i) compute the length and the number of shortest paths between all pairs, and ii) sum over all the pair-dependencies. For each node *i*, in the network, the number of "routing" paths to all other nodes (i.e., paths through which exist connectivity) going through *i* is counted, and this number determines the centrality *i*. The most common index is obtained by taking only the shortest paths as the routing paths. Formally the definition of betweenness centrality of a node *i*, is given by (1):

$$g(i) = \sum_{\{j,k\}} g_i(j,k). \tag{1}$$

where $\{j,k\}$ stands for summing each pair once, ignored the order, and $g_i(j,k)$ equals 1 if the shortest path between nodes j and k passes through node i, and 0 otherwise. In networks with no weight, i.e., where all edges have the same length, there may be more than one shortest path. In this case, it is common to take $g_i(j,k) = C_i(j,k)/C(j,k)$, where $C(j,k)$ is the number of shortest paths between j and k, and $C_i(j,k)$ is the number of those going through i. An improvement to this equation is presented in (3), where the numerator of the pair-dependency in (2), $C_i(j,k)$, is obtained by the Bellman criterion, $C(j,i) \cdot C(i,k)$, if the shortest paths between j and k pass through i [19]. A high centrality score indicates that a vertex can be reached by others on short paths, or that a vertex connects to others. In the methodology proposed, this metric is applied for all the non-terminal nodes of each fracture image, where the maximum value obtained is considered as the main node in the fracture system.

$$g(i) = \sum_{\mathrm{j}\neq\mathrm{i}\neq\mathrm{k}} \frac{C_i(j,k)}{C(j,k)}. \tag{2}$$

$$C_i(j,k) = \begin{cases} 0 & \text{if } C(j,k) < C(j,i) + C(i,k) \\ C(j,i) \cdot C(i,k) & \text{otherwise} \end{cases} \tag{3}$$

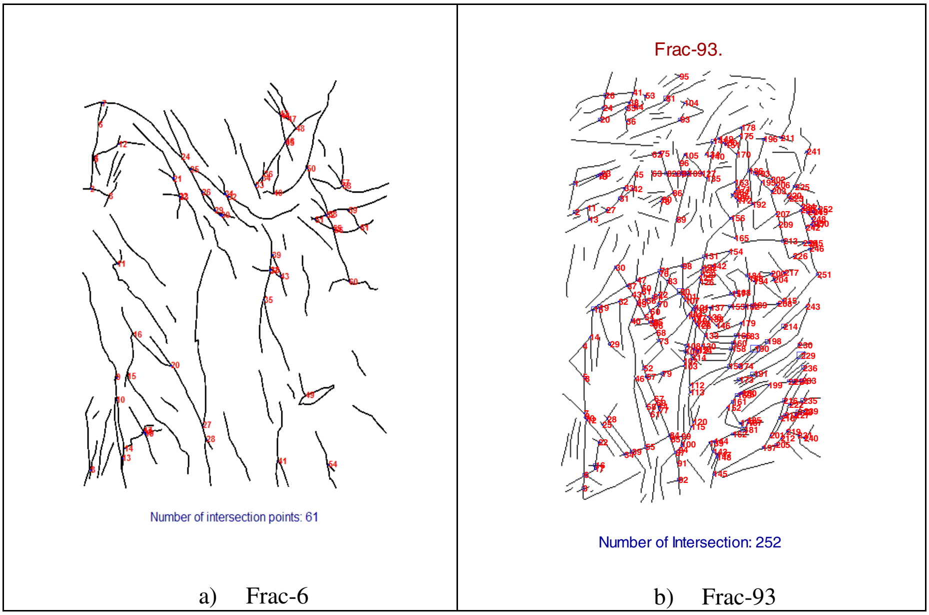
The second step in stage two consists in the computation of the intensity of non-terminal nodes that connect fractures. The idea is to carry out a sampling of the total image area with concentric circles centered at a strategic node, in this case, taking the most important node chosen by the betweenness centrality measure, and then to count the number of non-terminal nodes comprised in each circle. This process is described as follows. In Fig. 3 the intersection points among fractures are computed and presented, and the metric of centrality is done (see Fig. 4 and Fig. 5). Once the most important node in a graph is identified, it will be considered as the origin for the sampling circles. The third step is to draw the tendencies of all non-terminal nodes.

4 Results

In experiment, 100 hand-sample fracture images of the Gulf of Mexico oil Reservoir are considered. The images used were provided by experts of the geological area. Applying the first stage of the strategy (see Fig. 2), all the images are processed for recovering their fracture structure, in most cases required of the skeletonization process. For convenience only two images are shown in Fig. 1. Fig. 3 presents the resulting images after the application of the second step concerning to the identification of the non-terminal nodes or intersection points related to the images in Fig. 1, where the number of nodes is listed at the bottom of each image (61and 252 non-terminal nodes which correspond to Frac-6, and Frac-93, respectively). The next step is the determination of the topological structure of the fractured network employing the non-terminal nodes and the distance among their neighbors for its building. For drawing the resultant graph, the Cytoscape tool [20, 21] has been used, receiving as input data the adjacency matrix generated in the previous steps, this is carried out for each hand-sample image. The resultant graphs of Frac-6 and Frac-52 (see Fig. 4 and 5, respectively) are represented using a hierarchical structure. The number of conex components in Fract-6 is 17 and 30 conex components in Frac-93, which they represent the connectivity of the fractures studied.

In the second stage, for obtaining the main node, first the improved betweenness centrality measure is applied by using the Gephy tool [22, 23]. Then, the maximum value obtained using this metric is selected from all nodes in the fracture, and this maximum value of the corresponding node is considered the main node. To be precise a main node is defined as a node that has the highest connectivity according to the rest of the nodes in the fracture system, for this problem, the largest capability of transport fluids. This procedure was repeated for each fracture image. In Fig. 4 and 5, the main nodes are highlighted in black color, they correspond to nodes 7 and 25 in Frac-6, and node 76 in Frac-93; their betweenness centrality values are 195 and 17567.1, respectively. This operation was applied to the 100 test instances. In Table 1 only 20 fracture images are shown, where the first column presents the label of the image, the second column indicates the label of the main node obtained by applying the centrality metric, and the third column has the total number of non-terminal nodes analyzed. It is important to comment that some fractures obtained two main points, for example Frac-6 has 2 main nodes (7 and 25) since both have the same betweenness centrality measure.

Once the main node of each fracture is found, the next step is the determination of the distribution of the nodes in the whole network. In this experiment, we take the node or nodes identified as “main nodes” according to the definition given above, and use them as centers or concentric circles of increasing radius. The radii of the sampling circles is increased (arbitrarily) in 10 pixels, taking as origin the main node previously found. For instance, in Fig. 6a and 6b, the scaling of radius is shown, where the main nodes 25 and 76 were taken from Table 1 (corresponding to Frac-6 and Fract-93, respectively). Considering the node-25 of Frac-6, an example of growth of the number of non-terminal nodes in circular areas is presented in Table 2. The initial size of the radius is 10 pixels, and the next circles were generated in augment of 10 pixels, i.e., 20, 30, 40 pixels and so on until covering all the non-terminal nodes in the fracture image. In this instance, 84 regions were produced for covering the 61 nodes initiating in node-25 of Frac-6. The empirical cumulative distribution of the nodes of each image is shown in Fig. 7. In this figure each line represents the increase of the intersection points (non-terminal nodes) of each of the 100 fracture images, the axis of the coordinate is the distance of the radius in pixels, the axis of the abscise is the amount of the nodes found in each region, and the circle at the end of the line indicates the farthest node from the main node. Note in this figure that a general description from the set of 100 images two groups can be identified: one group where the intersection points of fractures (non-terminal nodes) are more scattered, and the second group where the intersection points are nearer among them.


Number of intersection points: 61
Frac-93.
Number of Intersection: 252

a) Frac-6

b) Frac-93

Fig. 3. Identification of intersection points (non-terminal nodes)

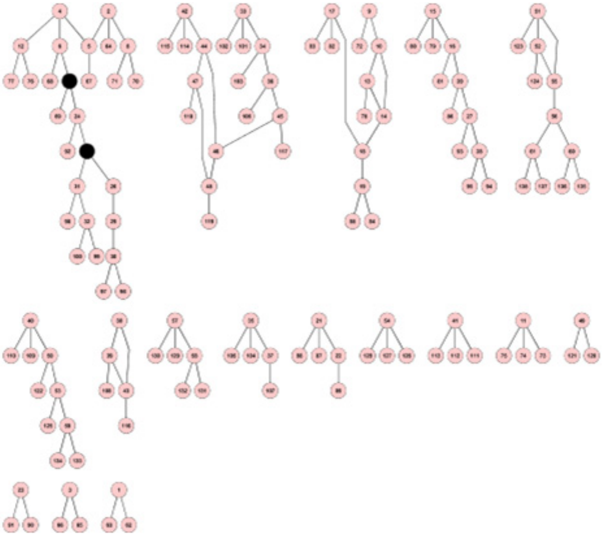


Fig. 4. Resultant graph of Frac-6

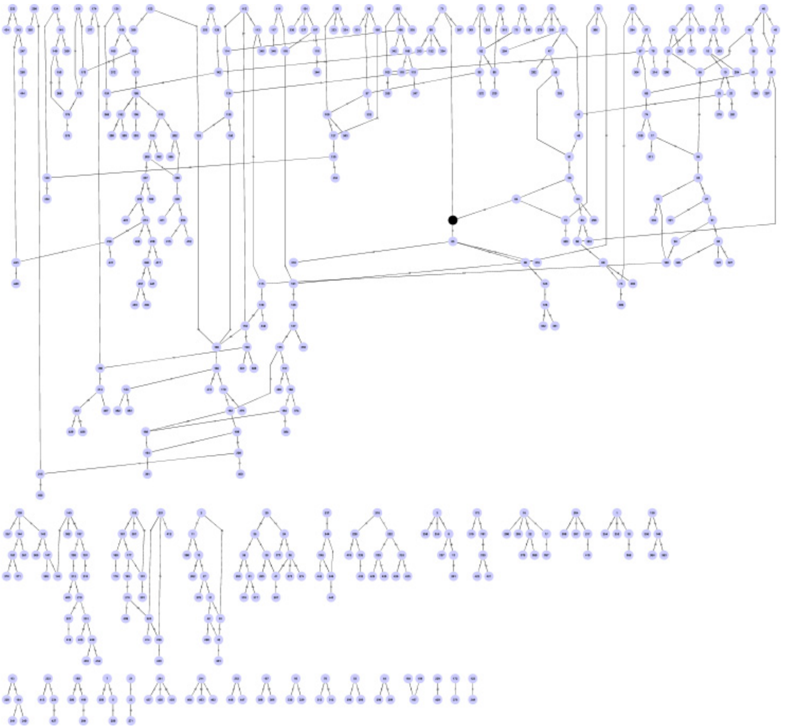


Fig. 5. Resultant graph of Frac-93

Table 1. Result of the betweenness centrality metric in the test set

Fracture label	Main node label	Total of non-terminal nodes	Fracture label	Main node label	Total of non-terminal nodes
Frac_(3)	node-15	24	Frac_(80)	node-13	107
Frac_(6)	node-7	61	Frac_(82)	node-28	64
Frac_(6)	node-25	61	Frac_(83)	node-47	107
Frac_(9)	node-34	69	Frac_(85)	node-50	143
Frac_(23)	node-19	28	Frac_(93)	node-76	252
Frac_(27)	node-18	43	Frac_(97)	node-8	29
Frac_(30)	node-18	36	Frac_(103)	node-137	158
Frac_(31)	node-52	79	Frac_(105)	node-47	79
Frac_(38)	node-176	244	Frac_(107)	node-20	34
Frac_(52)	node-32	50	Frac_(108)	node-34	51
Frac_(70)	node-90	113			

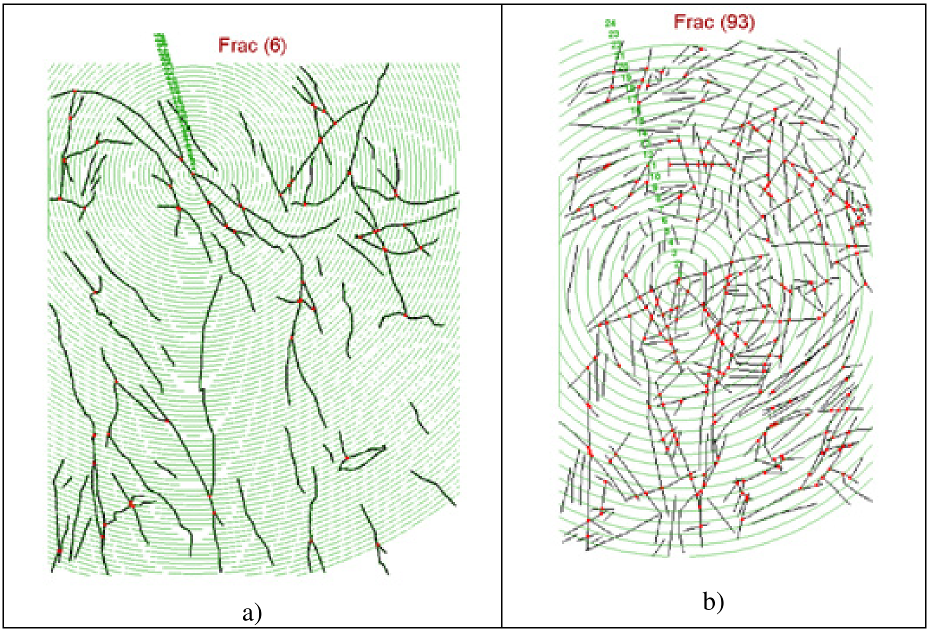


Fig. 6. Scaling of radius in a) node 25 of Frac-6 hand-sample, and b) node76 of Frac-93 hand-sample

Table 2. Radius generated from the main point 25 of Fig. 6.a

Radius length	No. of nodes	Radius length	No. of nodes	Radius length	No. of nodes	Radius length	No. of nodes
10	1	220	14	430	41	640	49
20	1	230	15	440	41	650	49
30	1	240	16	450	43	660	51
40	2	250	16	460	43	670	51
50	2	260	16	470	43	680	52
60	3	270	20	480	43	690	55
70	4	280	22	490	44	700	55
80	6	290	24	500	44	710	55
90	6	300	26	510	45	720	55
100	6	310	27	520	46	730	55
110	6	320	30	530	46	740	56
120	7	330	30	540	46	750	56
130	9	340	32	550	46	760	57
140	9	350	32	560	47	770	57
150	10	360	32	570	48	780	57
160	10	370	34	580	48	790	58
170	10	380	34	590	48	800	58
180	10	390	36	600	48	810	58
190	11	400	37	610	48	820	60
200	12	410	37	620	49	830	60
210	14	420	41	630	49	840	61

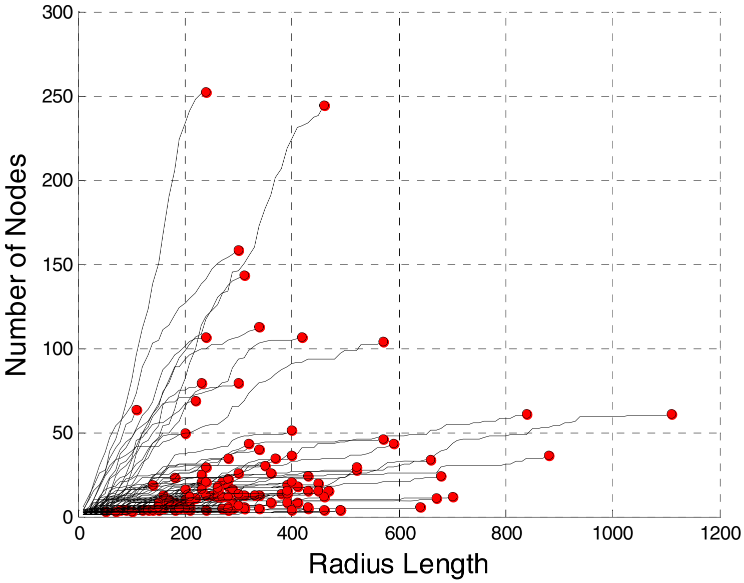


Fig. 7. Intensity of intersection points (non-terminal nodes) in different radius lengths

5 Conclusions and Future Work

In this work, a general methodology for analyzing the topology of fracture networks is presented. In our 100 hand-samples images two groups of fractures were identified. The first one characterizes fracture images with many nodes and short distances between them. These characteristics may indicate larger capability for fluid transport. The second group comprises fracture system with fewer nodes and large distances among them making the intersections sparser than in the previous group. We expect these results could be used as indicators in the evaluating in fluid flow capability. They can be compared with other geophysical features associated to the reservoir under study. This methodology can be applied to other kinds of problems modeled as graphs where the intersection of paths is important. As a possible line of future work, the analysis can be done from fractures obtained from RGB images, where the first stage of this methodology will be an important part in the recovery of the structure of the fractures.

On the other hand, it is convenient to explore other centrality metrics in complex networks that have emerged, such as the method named page rank [24], that is used to determine the importance of a node (used in the analysis of Web pages based on the links pointing to them). This algorithm initiates a random walk at a random node, following a random link at each node, with some small probability, at every step, of jumping to a randomly chosen node without following a link. This algorithm gives high importance (high probability of hitting) to nodes with a high number of links pointing to them, and also to nodes pointed to by these nodes.

References

1. Aguilera, R.: Naturally Fractured Reservoirs, 2nd edn. PennWel Books (1995)
2. Nelson, R.A.: Geological Analysis of Naturally Fractured Reservoir. Gulf Publ. (2001)
3. Narr, W., Schechter, D.S., Thompson, L.B.: Naturally Fractured Reservoir Characterization. Society of Petroleum Engineers, Richardson, Texas (2006)
4. Baker, R.O., Kuppe, F.: Reservoir Characterization for Naturally Fractured Reservoirs. Paper SPE 63286, SPE Annual Technical Conference and Exhibition, Dallas, Texas, October 1-4 (2000)
5. Gauthier, B.D.M., Garcia, M., Daniel, J.-M.: Integrated Fractured Reservoir Characterization: A Case Study in a North Africa Field. SPEREE 5(4), 284–294 (2002), SPE-79105-PA
6. Bogatkov, D., Babadagli, T.: Characterization of Fracture Network System of the Midale Field. Paper 2007-031, CIM 58th Annual Tech. Meet., Canadian International Petroleum Conf., Calgary, Canada, June 12-14 (2007)
7. Guerreiro, L., Silva, A.C., Alcobia, V., Soares, A.: Integrated Reservoir Charac-terisation of a Fractured Carbonate Reservoir. Paper SPE 58995, SPE International Petroleum Conference and Exhibition, Villahermosa, Mexico, February 1-3 (2000)
8. Warren, J.E., Price, H.S.: Flow in Heterogeneous Porous Media. Society of Petroleum Engineers. SPE Journal 1(3), 153–169 (1961)
9. Sarda, S., Jeannin, L., Basquet, R., Bourbiaux, B.: Hydraulic Characterization of Fractured Reservoirs: Simulation on Discrete Fracture Models. SPEREE 5(2), 154–162 (2002), SPE-77300-PA
10. Sarkar, S., Toksöz, M.N., Burns, D.R.: Fluid Flow Simulation in Fractured Reservoirs. Research Report, Earth Resources Laboratory, Department of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology, USA (2004)
11. Cacas, M.C., Ledoux, E., de Marsily, G., Tillie, B.: Modeling fracture flow with a stochastic discrete fracture network: calibration and validation. Water Resources Research 26, 479–489 (1990)
12. Newman, M.E.J.: Networks: An Introduction. Oxford University Press (2010)
13. Cohen, R., Havlin, S.: Complex Networks: Structures, Robustness and Function. Cambridge University Press (2010)
14. Witten, I.H., Frank, E., Hall, M.A.: Data Mining Practical Machine Learning Tools and Techniques, 3rd edn. (2011)
15. Bavelas, A.: A mathematical model for group structure. Applied Anthropology 7, 16–39 (1948)
16. Kolaczyk, E.D.: Statistical Analysis of Networks Data: Methods and Models. Springer Series in Statistics (2009)
17. Freeman Linton, C.: A set of Measures of Centrality Based on Betweenness. Sociometry 40(1), 35–41 (1977)
18. Anthonisse Jac, M.: The rush in a graph. Amsterdam. Mathematisch Centrum (mimeographed) (1971)
19. Brandes, U.: A Faster Algorithm for Betweenness Centrality. Journal of Mathematical Sociology 25(2), 163–177 (2001)
20. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research 13(11), 2498–2504 (2003)
21. Cytoscape Software, `http://www.cytoscape.org/`

22. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An Open Source Software for exploring and manipulating Networks. In: International AAAI Conference on Weblogs and Social Media (2009)
23. Gephi software, `http://gephi.org/`
24. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerange citation ranking: Bringing order to the web, Technical report, Stanford InfoLab (1999)
25. Yang, G., Myer, L.R., Brown, S.R., Cook, N.G.W.: Microscopic analysis of macroscopic transport properties of single natural fractures using graph theory algorithms. Geophys. Res. Lett. 22(11), 1429–1432 (1995), doi:10.1029/95GL01498.
26. Ghaffar, H.O., Nasseri, M.H.B., Young, R.P.: Fluid Flow Complexity in Fracture Network: Analysis with Graph Theory and LBM. arXiv:1107.4918 [cs.CE] (2012)
27. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining. In: American Association for Artificial Intelligence. AAAI Press (1996)

Comparative Study of Type-1 and Type-2 Fuzzy Systems for the Three-Tank Water Control Problem

Leticia Cervantes, Oscar Castillo, Patricia Melin, and Fevrier Valdez

Tijuana Institute of Technology
ocastillo@tectijuana.mx

Abstract. In this paper, simulation results with type-1 fuzzy systems and a type-2 fuzzy granular approach for intelligent control of non-linear dynamical plants are presented. First, the proposed method for intelligent control using a type-2 fuzzy granular approach is described. Then, the proposed method is illustrated with the benchmark case of three tank water level control. Finally, a comparison between a type-1 fuzzy system and the type-2 fuzzy granular system for water control is presented.

Keywords: Granular computing, Type-2 fuzzy logic, Fuzzy control.

1 Introduction

This paper focuses on the field of fuzzy logic, granular computing, type-1 and type-2 fuzzy systems and also considering the control area. In this work a comparative study of type-1 and type-2 fuzzy systems for the three-tank water control problem is presented. The comparison is illustrated with the problem of water level control for a three tank system. This control is carried out by controlling 5 valves whose outputs are the inputs to the 3 tanks. To control all the valves 5 fuzzy systems are used, each fuzzy system has to control one valve of the 3 tanks, after that the simulation is carry out and results are presented. Then a type-2 fuzzy system is used applying the concept of granular computing. Finally a comparative study is presented. The main contribution of the paper is showing that the type-2 fuzzy granular approach can work very well for a benchmark problem combining several individual controllers.

The rest of the paper is organized as follows: In section 2 some basic concepts to understand this work are presented, in section 3 the proposed method and problem description are presented, and finally conclusions are presented in section 4.

2 Background and Basic Concepts

In this section some basic concepts needed for this work are provided.

2.1 Granular Computing

Information granulation has emerged as one of the fundamental concepts of information processing giving rise to the discipline of Granular Computing. The concept itself

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 362–373, 2013.

permeates through a large variety of information systems. The underlying idea is intuitive and appeals to our commonsense reasoning. We perceive the world by structuring our knowledge, perceptions, and acquired evidence in terms of information granules-entities, which are abstractions of the complex word and phenomena. By being abstract constructs, information granules and their ensuing processing done under the umbrella of Granular Computing, provides a conceptual and algorithmic framework to deal with an array of decision-making, control, and prediction problems. Granular Computing supports human-centric processing, which becomes an inherent feature of intelligent systems. The required level of detail becomes conveniently controlled by making suitable adjustments to the size of information granules and their distribution in the space in which the problem at hand is being positioned and handled. In this sense, an inherent flexibility, which comes hand in hand with Granular Computing, can be effectively exploited in various ways. Three general tendencies encountered in Granular Computing can be identified: (a) a design of information granules of higher order, (b) a development of information granules of higher type, and (c) a formation of hybrid information granules [20].

The basic concepts of graduation and granulation form the core of fuzzy logic, and are the main distinguishing features of fuzzy logic. More specifically, in fuzzy logic everything is or is allowed to be graduated, i.e., be a matter of degree or, equivalently, fuzzy. Furthermore, in fuzzy logic everything is or is allowed to be granulated, with a granule being a clump of attribute values drawn together by in distinguishability, similarity, proximity, or functionality. The concept of a generalized constraint serves to treat a granule as an object of computation. Graduated granulation, or equivalently fuzzy granulation, is a unique feature of fuzzy logic. Graduated granulation is inspired by the way in which humans deal with complexity and imprecision. The concepts of graduation, granulation, and graduated granulation play key roles in granular computing. Graduated granulation underlies the concept of a linguistic variable, i.e., a variable whose values are words rather than numbers. In retrospect, this concept, in combination with the associated concept of a fuzzy if–then rule, may be viewed as a first step toward granular computing [21][1][24][25]. Granular Computing (GrC) is a general computation theory for effectively using granules such as subsets, neighborhoods, ordered subsets, relations (subsets of products), fuzzy sets (membership functions), variables (measurable functions), Turing machines (algorithms), and intervals to build an efficient computational model for complex with huge amounts of data, information and knowledge. Fuzzy and rough set theories, neutrosophic computing, quotient space, belief functions, machine learning, databases, data mining, cluster analysis, interval computing, more recently social computing, all involve granular computing. The topics and areas include: Computational Intelligence, in this area we have sub-areas such as: Neural Networks, Fuzzy Systems, Evolutionary Computation, Rough Sets and Formal Concept Analysis, etc. In the area of data mining and learning theory there are sub-areas such as: Stochastic learning, Machine learning, Kernel Machines, etc. In applications it can be used in bioinformatics, Medical Informatics and Chemical Informatics, e-Intelligence, Web Intelligence, Web Informatics, Web Mining and Semantic Web [10][2].

2.2 Type-2 Fuzzy Logic

A fuzzy system is a system that uses a collection of membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy system are usually of a form similar to the following: if *x* is low and *y* is high then *z* = medium, where *x* and *y* are input variables (names for known data values), *z* is an output variable (a name for a data value to be computed), low is a membership function (fuzzy subset) defined on *x*, high is a membership function defined on *y*, and medium is a membership function defined on *z*. The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. A type-2 fuzzy system is similar to its type-1 counterpart, the major difference being that at least one of the fuzzy sets in the rule base is a Type-2 Fuzzy Set. Hence, the outputs of the inference engine are Type-2 Fuzzy Sets, and a type-reducer is needed to convert them into a Type-1 Fuzzy Set before defuzzification can be carried out. An example of a Type-2 Fuzzy Set $\tilde{X}_{mn}$ is shown in Fig. 1.

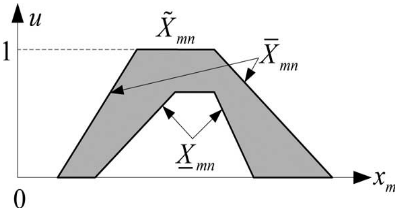


Fig. 1. Type-2 fuzzy set

Its upper membership function (UMF) is denoted $\overline{X}_{mn}$ and its lower membership function (LMF) is denoted $\underline{X}_{mn}$. A Type-2 fuzzy logic system has *M* inputs *{xm} m*=1,2,...,*M* and one output *y*. Assume the *m*th input has *Nm* MFs in its universe of discourse $\mathbb{X}_m$ Denote the *n*th MF in the *m*th input domain as $\tilde{X}_{mn}$. A complete rule-base with all possible combinations of the input fuzzy system consists of $K = \prod_{m=1}^{M} N_m$rules in the form of:

$$\tilde{R}^k\text{: IF } x_1 \text{ is } \tilde{X}_{1,n_{1k}} \text{ and ...and } x_M \text{ is } \tilde{X}_{M,n_{Mk}} \quad (1)$$
$$[\underline{y}_k, \overline{y}_k],\ n_{ik} = 1,2,\ldots,N_i,\ k = 1,2,\ldots,K$$

where $[\underline{y}_k, \overline{y}_k]$is a constant interval, and generally, it is different for different rules. $[\underline{y}_k, \overline{y}_k]$ represents the centroid of the consequent Type-2 Fuzzy Set of the *k*th rule. When $\underline{y}_k = \overline{y}_k$, this rulebase represents the simplest TSK model, where each rule consequent is represented by a crisp number. Again, this rulebase represents the most commonly used Type-2 Fuzzy Logic System in practice. When KM type-reduction and center-of-sets defuzzification are used, the output of a Type-2 Fuzzy Logic System with the aforementioned structure for an input x = (x_1, x_2, . . . , x_m) is computed as:

$$y(\mathbf{x}) = \frac{y_l(\mathbf{x}) + y_r(\mathbf{x})}{2} \tag{2}$$

Where:

$$\begin{aligned} y_l(\mathbf{x}) &= \min_{\forall f_k \in [\underline{f}_k, \overline{f}_k]} \frac{\sum_{k=1}^{K} f_k \underline{y}_k}{\sum_{k=1}^{K} f_k} \\ &= \frac{\sum_{k=1}^{k_l} \overline{f}_k \underline{y}_k + \sum_{k=k_l+1}^{K} \underline{f}_k \underline{y}_k}{\sum_{k=1}^{k_l} \overline{f}_k + \sum_{k=k_l+1}^{K} \underline{f}_k} \end{aligned} \tag{3}$$

$$\begin{aligned} y_r(\mathbf{x}) &= \max_{\forall f_k \in [\underline{f}_k, \overline{f}_k]} \frac{\sum_{k=1}^{K} f_k \underline{y}_k}{\sum_{k=1}^{K} f_k} \\ &= \frac{\sum_{k=1}^{k_r} \underline{f}_k \overline{y}_k + \sum_{k=k_r+1}^{K} \overline{f}_k \overline{y}_k}{\sum_{k=1}^{k_r} \underline{f}_k + \sum_{k=k_r+1}^{K} \overline{f}_k} \end{aligned} \tag{4}$$

in which $[\underline{y}_k, \overline{y}_k]$ is the firing interval of the *k*th rule, i.e.

$$\begin{aligned} \underline{f}_k &= \mu_{\underline{X}_{1,n_{1k}}}(x_1) \star \mu_{\underline{X}_{2,n_{2k}}}(x_2) \star \cdots \star \mu_{\underline{X}_{M,n_{Mk}}}(x_M) \\ \overline{f}_k &= \mu_{\overline{X}_{1,n_{1k}}}(x_1) \star \mu_{\overline{X}_{2,n_{2k}}}(x_2) \star \cdots \star \mu_{\overline{X}_{M,n_{Mk}}}(x_M). \end{aligned} \tag{5}$$

Observe that both $\underline{f}_k$ *k* and $\overline{f}_k$ are continuous functions when all Type-2 Membership Functions are continuous. A Type-2 Fuzzy System $\tilde{X}$ is continuous if and only if both its UMF and its LMF are continuous Type-1 Fuzzy Systems [11][13][16] [12][6][5][4].

3 Description of the Problem

There are several methods available to solve control problems such as: PID control, fuzzy control and some other methods using neural networks [14][22][18][24][19][17][14][15][8][9]. In this work the main goal is to make a comparative study of type-1 and type-2 fuzzy systems in automatic control of nonlinear dynamic plants using a fuzzy granular approach and also to implement a granular architecture in a particular control problem, which is the control of 3 water tank system, and also to use a simulation tool to test the fuzzy controller.

The 3 tanks include valves that are open or closed; these valves must be well controlled to give the desired level of water in each of the three tanks. The end tanks have a valve that fills and in the middle of the 3 tanks there are two valves that control the water level between tanks 1 and 2, and tanks 2 and 3. The water tank 3 has a valve to output more water flow. The problem can be reflected in Fig.2.

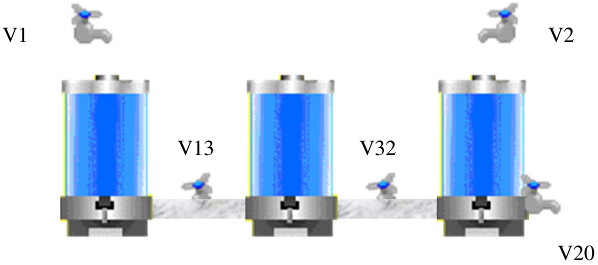
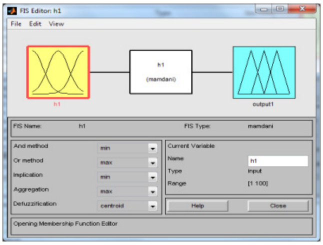


Fig. 2. Water control of 3 tanks

3.1 Type-1 Fuzzy System

Type-1 fuzzy systems can be used to control the water level, and five fuzzy systems are used to carry out the total control. Each fuzzy system is used to control one valve. The five systems are shown in Fig. 3 to 7.



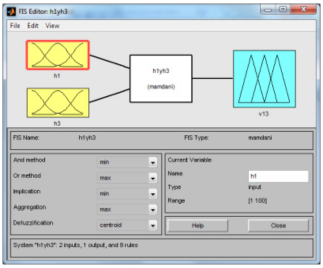
FIS Name: h1 FIS Type: mamdani

And method min
Or method max
Implication min
Aggregation max
Defuzzification centroid

Current Variable
Name h1
Type input
Range [1 100]

Opening Membership Function Editor

Fig. 3. Fuzzy system to control valve 1



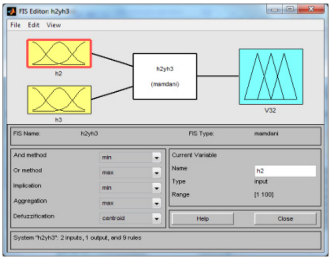
FIS Name: h1yh3 FIS Type: mamdani

And method min
Or method max
Implication min
Aggregation max
Defuzzification centroid

Current Variable
Name h1
Type input
Range [1 100]

System "h1yh3": 2 inputs, 1 output, and 9 rules

Fig. 4. Fuzzy system to control valve 13



FIS Name: h2yh3 FIS Type: mamdani

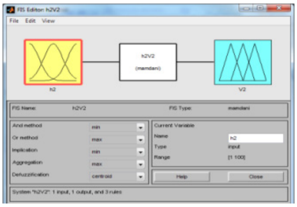
And method min
Or method max
Implication min
Aggregation max
Defuzzification centroid

Current Variable
Name h2
Type input
Range [1 100]

System "h2yh3": 2 inputs, 1 output, and 9 rules

Fig. 5. Fuzzy system to control valve 32

This problem has 2 valves between the tanks and it is necessary to use two inputs (tank1 and tank2 or tank 2 and tank3).



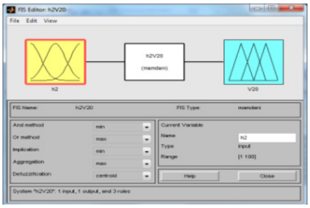
FIS Name: h2V2 FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: h2
Type: input
Range: [1 100]

System "h2V2": 1 input, 1 output, and 3 rules

Fig. 6. Fuzzy system to control valve 2



FIS Name: h2V20 FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: h2
Type: input
Range: [1 100]

System "h2V20": 1 input, 1 output, and 3 rules

Fig. 7. Fuzzy system to control valve 20

After creating the fuzzy systems the simulation was carry out in the Matlab programming language. The simulation plant that was used is shown in Fig.8.

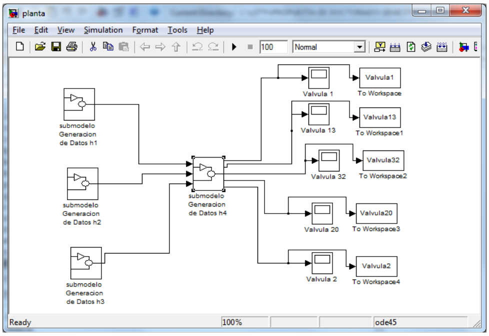


Fig. 8. Simulation Plant

The simulation was carried out with different types of membership functions. Results are shown in section 4.

3.2 Type-2 Fuzzy System

A type-2 fuzzy system with granular computing is used to control the water level, after using type-1 fuzzy system in the simulation plant, the outputs of the type-1 was use as the new inputs for type-2 and this is done to obtain new outputs and to have a better result. The results with this type-2 granular fuzzy systems are shown later in section 4. The simulation plant is shown in Fig. 9.

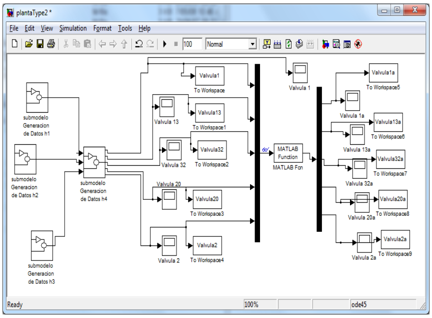
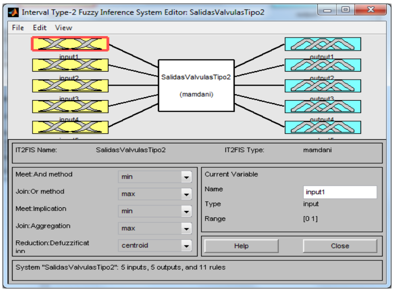


Fig. 9. Simulation plant

The granular type-2 fuzzy system used in this work is shown in Fig. 10. The inputs in type-2 fuzzy system were obtained from the outputs of the type-1 fuzzy system, and the outputs of type-2 fuzzy system are new outputs to have a better control.



IT2FIS Name: SalidasValvulasTipo2
IT2FIS Type: mamdani

Meet:And method: min
Join:Or method: max
Meet:Implication: min
Join:Aggregation: max
Reduction:Defuzzification: centroid

Current Variable
Name: input1
Type: input
Range: [0 1]

System "SalidasValvulasTipo2": 5 inputs, 5 outputs, and 11 rules

Fig. 10. Type-2 fuzzy system

The simulation was carried out using a type-2 fuzzy system and the results are shown in section 4. The rules used in the type-2 fuzzy system are different from the rules used in the type-1 fuzzy system, the number of the rules depend on the number of the input, membership functions and output. The comparison was perform first establishing a high level of water in each tank, once having established this level simulation is performed and the valves begin to open and close up to the desired level. And the error is the difference with the desired water level and the water level obtained. The type of membership functions that were used are triangular type.

4 Comparative Studies of Type-1 and Type-2 Fuzzy Control

Working with type-1 and type-2 fuzzy systems can be useful because it would provide an idea of which fuzzy system is better to use in each problem. Is important to know that not in all cases type-2 is better, but is necessary to study the problem and to do tests with type-2 and type-1 fuzzy systems.

Observe the behavior with both types of fuzzy systems and compare the results is very important. In this case, this work considers type-1 and type-2 fuzzy systems for water level control. The results that were obtained with a type -1 fuzzy system are shown in Table 1 and the results with a type-2 fuzzy system are shown in Table 2.

Table 1. Results for simulation plant with a type-1 fuzzy controller

Valve 1	Valve 13	Valve 2	Valve 20	Valve 32
0.7864	0.4743	0.5239	0.5215	0.6343
0.5959	0.6128	0.7136	0.4718	0.7965
0.8694	0.5102	0.645	0.5892	0.6486
1.0447	0.9318	0.823	0.823	0.7976
0.6887	0.8085	0.5598	0.6192	0.4842
0.5573	0.8079	0.5575	0.7933	0.4118
0.6667	0.4615	0.7084	0.5142	0.5311
0.6428	0.8186	0.9575	0.3879	0.3393
0.8247	0.6718	1.0955	0.8056	0.4376
1.0525	0.8530	1.098	0.5476	0.6292
0.8042	0.8979	1.0844	0.5684	0.3255
0.4494	0.9520	1.0289	0.7521	0.7745
0.7780	0.6679	0.671	0.6341	0.2257
0.4021	0.6618	0.7973	0.7774	0.6009
0.6879	0.7449	0.9034	0.462	0.6124
0.6432	0.7069	0.7258	1.096	0.5172
0.7590	0.6879	0.7044	0.8517	0.5216
1.0404	0.8798	0.8758	0.7772	0.554
0.8343	0.6014	0.7078	0.6553	0.9855
0.9223	0.6721	0.7381	0.5878	0.6048
0.6786	0.5999	0.5936	0.5943	0.3973
0.4929	0.8258	0.4361	0.6293	0.4468
0.8291	0.6187	0.6133	0.4118	0.6703
0.7849	0.6596	0.6789	0.6309	0.6776
0.7407	0.5966	0.5705	0.6035	0.5415
0.8589	0.9600	0.5934	0.8404	0.5572
0.8999	0.6375	0.6852	0.6966	0.5304
0.8747	0.5267	0.4607	0.4643	0.3716
0.6249	0.6306	0.6153	0.433	0.4844
0.8096	0.6108	0.6969	0.4499	0.9614

Table 2. Results for simulation plant with a type-2 fuzzy

Valve 1	Valve 13	Valve 2	Valve 20	Valve 32
0.0185	0.0187	0.0187	0.1248	0.1881
0.0185	0.0187	0.0187	0.1248	0.1880
0.0147	0.0172	0.0172	0.1145	0.2137
0.0183	0.0168	0.0168	0.1122	0.2196
0.0181	0.0185	0.0185	0.1233	0.1918
0.0172	0.0177	0.0177	0.1178	0.2054
0.0164	0.0174	0.0174	0.1157	0.2107
0.0174	0.0175	0.0175	0.1169	0.2076
0.0153	0.0174	0.0174	0.1162	0.2094
0.0185	0.0184	0.0184	0.1228	0.1929
0.0181	0.0188	0.0188	0.1255	0.1861
0.0174	0.0206	0.0206	0.1375	0.1561
0.0177	0.0185	0.0185	0.1231	0.1921
0.0142	0.018	0.018	0.1199	0.2003
0.0159	0.0152	0.0152	0.1015	0.2463
0.0166	0.0175	0.0175	0.117	0.2075
0.0150	0.0152	0.0152	0.1012	0.2471
0.0149	0.0165	0.0165	0.1102	0.2244
0.0116	0.0193	0.0193	0.1284	0.1789
0.0145	0.0163	0.0163	0.1085	0.2289
0.0165	0.0167	0.0167	0.1114	0.2216
0.0175	0.0187	0.0188	0.1249	0.1877
0.0127	0.0182	0.0182	0.1215	0.1963
0.0168	0.0181	0.0181	0.1203	0.1991
0.0161	0.0164	0.0164	0.1094	0.2264
0.0193	0.0199	0.0199	0.1327	0.1681
0.0143	0.016	0.016	0.1064	0.2339
0.017	0.0175	0.0175	0.1165	0.2088
0.0157	0.0156	0.0156	0.104	0.2400
0.0169	0.0150	0.0150	0.1000	0.2500

Having obtained the above results, the t student tests were performed, and the results are shown in Tables 3 to 7.

Table 3. Results for the t student test using triangular membership function in type-1 and type-2 fuzzy systems in valve 1

	N	Mean	Deviation std.	Mean of std. error
Type-1 fuzzy system	30	0.755	0.164	0.030
Type-2 fuzzy system	30	0.01639	0.00182	0.00033

T = 24.64 P = 0.000 GL = 58.

Table 4. Results for the t student test using triangular membership function in type-1 and type-2 fuzzy systems in valve 13

	N	Mean	Deviation std.	Mean of std. error
Type-1 fuzzy system	30	0.703	0.141	0.026
Type-2 fuzzy system	30	0.01754	0.00139	0.00025

T = 26.64 P = 0.000 GL = 58.

Table 5. Results for the t student test using triangular membership function in type-1 and type-2 fuzzy systems in valve 2

	N	Mean	Deviation std.	Mean of std. error
Type-1 fuzzy system	30	0.729	0.183	0.033
Type-2 fuzzy system	30	0.01755	0.00139	0.00025

T = 21.32 P = 0.000 GL = 58.

Table 6. Results for the t student test using triangular membership function in type-1 and type-2 fuzzy systems in valve 20

	N	Mean	Deviation std.	Mean of std. error
Type-1 fuzzy system	30	0.633	0.163	0.030
Type-2 fuzzy system	30	0.011696	0.00929	0.0017

T = 17.34 P = 0.000 GL = 58.

Table 7. Results for the t student test using triangular membership function in type-1 and type-2 fuzzy systems in valve 32

	N	Mean	Deviation std.	Mean of std. error
Type-1 fuzzy system	30	0.569	0.175	0.032
Type-2 fuzzy system	30	0.2076	0.0233	0.0042

T = 11.19 P = 0.000 GL = 58.

Table 3 shows, that there is statistical evidence to say that there is a significant difference in the results presented in this paper with regards to type-2 (more than 95% confidence). In other words, the type-2 fuzzy system with granular computing generated a significant increase in the controller to have better control. Tables 4, 5, 6 and 7 also show comparisons between the same types of membership function, but each table compared each valve behavior and we can say that significant differences were also observed with a margin of more than 95% confidence.

5 Conclusions

Based on the results obtained with the type-1 and type-2 fuzzy systems it can be said that to achieve control of the water level, the fuzzy controller is a good alternative to achieve good control. The obtained results that were better that when a type-2 fuzzy system with granular computing was used. When considering a complex control problem is important to work with a type-1 fuzzy system and to observe in this case the valve behavior and then use a type-2 fuzzy system to improve the level water control. In this work a comparison between type-1 fuzzy system and type-2 fuzzy system was present in case of the study of the 3 water control tanks, the comparison can be realize in many problem to improve the results [3].

References

1. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordercht (2003)
2. Bargiela, A., Pedrycz, W.: The roots of Granular Computing. In: GrC 2006, pp. 806–809. IEEE (2006)
3. Castillo, O., Martinez-Marroquín, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Information Sciences 192, 19–38 (2012)
4. Castro, J.R., Castillo, O., Melin, P., Rodríguez-Díaz, A.: Building Fuzzy Inference Systems with a New Interval Type-2 Fuzzy Logic Toolbox. In: Gavrilova, M.L., Tan, C.J.K. (eds.) Trans. on Comput. Sci. I. LNCS, vol. 4750, pp. 104–114. Springer, Heidelberg (2008)

5. Castro, J.R., Castillo, O., Melin, P., Rodriguez, A., Mendoza, O.: Universal Approximation of a Class of Interval Type-2 Fuzzy Neural Networks Illustrated with the Case of Nonlinear Identification. In: IFSA/EUSFLAT Conf., pp. 1382–1387 (2009)
6. Castro, J.R.: Tutorial type-2 fuzzy logic theory and applications. In: 3rd International Seminar on Computational Intelligence. IEEE Computational Intelligence Society, Mexico (2006)
7. Cervantes, L., Castillo, O.: Design of a Fuzzy System for the Longitudinal Control of an F-14 Airplane. SCI, vol. 318, pp. 213–224 (2011)
8. Cervantes, L., Castillo, O.: Hierarchical Genetic Algorithms for Optimal Type-2 Fuzzy System Design. In: Annual Meeting of the North American Fuzzy Information Processing Society, pp. 324–329 (2011)
9. Cervantes, L., Castillo, O.: Automatic Design of Fuzzy Systems for Control of Aircraft Dynamic Systems with Genetic Optimization. In: World Congress and AFSS International Conference, pp. OS-413-1–OS-413-7 (2011)
10. Computer Society, The 2011 IEEE Internaional Confenrece on Granular Computing Sapporo, Japan, August 11-13, 2011. IEEE GrC 2011 Indexed by EI (GrC 2005, 2006, 2007, 2008, 2009 are indexed by EI). Call for Papers
11. Dongrui, W., Mendel, J.: On the Continuity of Type-1 and Interval Type-2 Fuzzy Logic Systems. IEEE T. Fuzzy Systems 19(1), 179–192 (2011)
12. Dongrui, W.: A Brief Tutorial on Interval Type-2 Fuzzy Sets and Systems (July 22, 2010)
13. Karnik, N., Mendel, J.: Centroid of a type-2 fuzzy set. Information Sciences 132, 195–220 (2001)
14. Melin, P., Castillo, O.: Intelligent control of aircraft dynamic systems with a new hybrid neuro- fuzzy–fractal Approach. Journal 142(1), 161–175 (2002)
15. Melin, P., Castillo, O.: Adaptive intelligent control of aircraft system with a hybrid approach combining neural network, fuzzy logic and fractal theory. Appl. Soft Comput. 3(4), 363–378 (2003)
16. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
17. Niemann, H., Stoustrup, J.: Passive fault tolerant control of a double inverted pendulum a case study. Control Engineering Practice 13(8), 1047–1059 (2005)
18. No, T.S., Mina, B.M., Stone, R.H., Wong, K.C.: Control and simulation of arbitrary flight trajectory-tracking. Department of Aerospace Engineering, Chonbuk National University, Deokjin Dong, Chonju NSW, pp. 560–756 (2006)
19. No, T.S., Kim, J.E., Moon, J.H., Kim, S.J.: Modeling, control, and simulation of dual rotor wind turbine generator system. Renewable Energy 34(10), 2124–2132 (2009)
20. Pedrycz, W., Chen, S.: Granular Computing and Intelligent Systems. Springer, Heidelberg (2011) ISBN 978-3-642-19819-9
21. Pedrycz, W., Skowron, A., Kreinovich, V.: Handbook granular computing. Wiley-Interscience, New York (2008) ISBN:0470035544 9780470035542 2008
22. Sepulveda, R., Castillo, O., Melin, P., Montiel, O.: An Efficient Computational Method to Implement Type-2 Fuzzy Logic in Control Application. In: Melin, P., Castillo, O., Ramírez, E.G., Kacprzyk, J., Pedrycz, W. (eds.) Anal. and Des. of Intell. Sys. using SC Tech. ASC, vol. 41, pp. 45–52. Springer, Heidelberg (2007)
23. Yu, Y., Chen, L., Sun, F., Wu, C.: Matlab/Simulink-based simulation for digital-control system of marine three-shaft gas-turbine. Applied Energy 80(1), 1–10 (2005)
24. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Comput. 2, 23–25 (1998)
25. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans.Syst. Man Cybern. SMC-3, 28–44 (1973)

Parallel Particle Swarm Optimization with Parameters Adaptation Using Fuzzy Logic

Fevrier Valdez, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology, Tijuana BC, México

Abstract. We describe in this paper a Parallel Particle Swarm Optimization (PPSO) method with dynamic parameter adaptation to optimize complex mathematical functions. Fuzzy Logic is used to adapt the parameters of the PSO in the best way possible. The PPSO is shown to be superior to the individual evolutionary methods on the set of benchmark functions.

Keywords: Parallel FPSO, FGA, GA.

1 Introduction

We describe in this paper a parallel particle swarm optimization method combining different topologies, to give us an improved PSO method. We apply the method to mathematical function optimization to validate the approach. Also, in this paper the comparison of a Genetic Algorithm (GA) [1] and Particle Swarm Optimization (PSO) [2] for the optimization of mathematical functions is considered. In this case, we are using a set of mathematical benchmark functions [5][13][15] to compare the optimization results between a GA, PSO and PPSO.

The main motivation of this method is to combine the different topologies of PSO in parallel form to achieve best results very fast. The topologies considered in this paper are the following: Star, Ring, Wheel and Pyramid. We are using a fuzzy logic system to perform dynamical parameter adaptation. The fuzzy system is used to adapt the parameters on PPSO, in this case, the inertia weight, and this way obtaining the best solution to the optimization problem. The main goal of the fuzzy system is to evaluate the outputs of PPSO in each generation and change if is necessary some important parameters. The criterion for stopping the method is the maximum number of generations.

The paper is organized as follows: in section 2 a description about Genetic Algorithms for optimization problems is presented, in section 3 the Particle Swarm Optimization is presented, in section 4 the proposed method PPSO is described, in section 5 we can appreciate the mathematical functions that were used for this research, in section 6 the simulations results are described, in section 7 we can appreciate a comparison between GA, PSO and PPSO, in section 8 the conclusions obtained after the study of the proposed methods studied are presented.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 374–385, 2012.

2 Genetic Algorithms for Optimization

Holland, from the University of Michigan initiated his work on genetic algorithms at the beginning of the 1960s. His first achievement was the publication of *Adaptation in Natural and Artificial System* [7] in 1975.

He had two goals in mind: to improve the understanding of natural adaptation process, and to design artificial systems having properties similar to natural systems [8].

The basic idea is as follows: the genetic pool of a given population potentially contains the solution, or a better solution, to a given adaptive problem. This solution is not "active" because the genetic combination on which it relies is split between several subjects. Only the association of different genomes can lead to the solution.

Holland's method is especially effective because it not only considers the role of mutation, but it also uses genetic recombination, (crossover) [9]. The crossover of partial solutions greatly improves the capability of the algorithm to approach, and eventually find, the optimal solution.

The essence of the GA in both theoretical and practical domains has been well demonstrated [1]. The concept of applying a GA to solve engineering problems is feasible and sound. However, despite the distinct advantages of a GA for solving complicated, constrained and multiobjective functions where other techniques may have failed, the full power of the GA in application is yet to be exploited [12] [14].

In Figure 1 we show the reproduction cycle of the Genetic Algorithm.

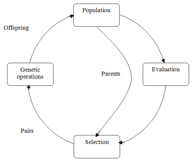


Fig. 1. The Reproduction cycle

The Simple Genetic Algorithm can be expressed in pseudo code with the following cycle:

```
1. Generate the initial population of individuals aleatorily P(0).
2. While (number _ generations <= maximum _ numbers _ generations)
     Do:
       {
         Evaluation;
         Selection;
         Reproduction;
         Generation ++;
       }
3. Show results
   4. End
```

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy in 1995, inspired by the social behavior of bird flocking or fish schooling [3].

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA) [6]. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, the PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [10].

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far (The fitness value is also stored). This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p*best* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations.

In the past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods [11].

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

The pseudo code of the PSO is as follows

```
 For each particle
  Initialize particle
End
Do
  For each particle
    Calculate fitness value
    If the fitness value is better than the best fitness value (pBest) in history
      set current value as the new pBest
  End
  Choose the particle with the best fitness value of all the particles as the gBest
  For each particle
    Calculate particle velocity
    Update particle position
  End
While maximum iterations or minimum error criteria is not attained
```

4 Proposed Architecture of Parallel Particle Swarm Optimization

The general approach of the proposed PPSO method can be seen in Figure 2. The method can be described as follows:

1. The swarm and parameters are initialized.
2. The position and velocity are set randomly
3. It receives a mathematical function to be optimized
4. It evaluates with each topology on each core.
5. The position and velocity are updated.
6. After each particle on each topology is evaluated.
7. Follow step the best solutions of each topology is obtained and compared to find the global best.
8. In the topology with bad performance a fuzzy system is used to adapt the inertia weight to improve the results in the next iterations.
9. A main fuzzy system is responsible for receiving values resulting.
10. The main fuzzy system decides which topology of PSO is best (ring, star, wheel or pyramid)
11. Repeat the above steps until the termination criterion of the algorithm is met.

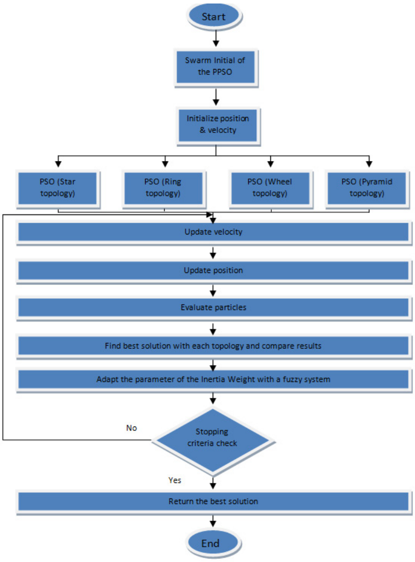


Fig. 2. The PPSO scheme

The basic idea of the PPSO scheme is to combine the advantages of each topology executed on 1 core where each PSO is using a fuzzy system for parameter adaptation.

The fuzzy system is of Mamdani type because it is more common in this type of fuzzy control and the defuzzification method was the centroid. In this case, we are using this type of defuzzification because in other papers we have achieved good results [5]. The membership functions are of triangular form in the inputs and outputs. Also, the membership functions were chosen of triangular form based on past experiences in this type of fuzzy control. The fuzzy system consists of 9 rules. For example, one rule is if error is P and DError is P then best value is P. The other two fuzzy systems are similar to the main fuzzy system.

5 Benchmark Mathematical Functions

To validate our method we used a set of 5 benchmark mathematical functions, called F1, F2, F3, F4 and F5; all functions were evaluated with different numbers of variables. In this case, the simulation results were obtained with 8, 16, 32, 64 and 128

The F1 function is defined by this equation:

$$f_1 = \sum_{n=i}^{N} x_n^2 \quad \text{minimum} = 0, \text{ for } -\infty \le x \le \infty \tag{1}$$

The F2 function is defined by this equation:

$$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2 \quad \text{minimum} = 0, \text{ for } -65.536 \le xi \le 65.536 \tag{2}$$

The F3 function is defined by this equation:

$$f(x) = \sum_{i=1}^{n-1} 100.\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2 \tag{3}$$

$$\text{minimum=0, for} -2.048 \le xi \le 2.048$$

Equation 4 shows the description to F4 function.

$$10N + \sum_{n=1}^{N} \left[x_n^2 - 10\cos\left(2\pi x_n\right) \right] \tag{4}$$

$$\text{minimum} = 0, \text{ for } -\infty \le x \le \infty$$

Equation 5 shows the description to F5 function.

$$1 + \sum_{n=1}^{N} \frac{x_n^2}{4000} \prod_{n=1}^{N} \cos\left(x_n\right) \tag{5}$$

$$\text{minimum} = 0, \text{ for } -\infty \le x \le \infty$$

6 Simulations Results

Several tests of the PPSO, PSO and GA algorithms were made with an implementation in the Matlab programming language.

The implementation was developed using a computer with a processor Intel Core 2 Quad of 64 bits that works to a frequency of clock of 2.5 GHz, 6 GB of RAM Memory and Ubuntu Linux Operating System. The results obtained after applying the GA to the mathematical functions are shown in Tables 1, 2, 3, 4 and 5. The parameters used in the Tables are:

- F1, F2, F3, F4 and F5 = Name of function.
- VARIABLES= Number of variables used to evaluation.
- BEST= The best result obtained.
- AVERAGE= The average of 50 times.
- WORST= The worst result obtained.

The parameters used on all tests with the GA were: population size = 100 individuals

- Crossover (k1) = 80%,
- Mutation (k2) = 5%,
- Selection= roulette.
- On PSO the parameters were: swarm size= 100 particles,
- Cognitive acceleration (c1) = 1,
- Social acceleration (c2) = 0.5,
- Value of velocity at the beginning = 0.95, constriction factor = 1.
- On PPSO the parameters were different for W (inertia weight), the parameter was fuzzy, because were obtained dynamically when the method is running, the fuzzy system can be obtaining this parameter the best way for achieve good results. The four topologies (ring, star, wheel and pyramid) were used with PPSO.

6.1 Simulation Results with the Genetic Algorithm (GA)

From Table 1 it can be appreciated that after executing the GA 50 times, with different number of variables, we can note the best, average and worst results for the F1 function.

Table 1. Experimental results with ga for f1 function

VARIABLES	BEST	AVERAGE	WORST
8	8.66E-07	0.00094	0.0070
16	4.09E-06	0.00086	0.0083
32	1.14E-06	0.00094	0.0056
64	1.00E-05	0.00098	0.0119
128	1.00E-05	9.42E-04	0.0071

From Table 2 it can be appreciated that after executing the GA 50 times, with different number of variables, we can see the best, average and worst results for the F2 function.

Table 2. Experimental results with ga for f2 function

VARIABLES	BEST	AVERAGE	WORST
8	5.29E-05	0.05823	0.30973
16	0.00071	0.05683	0.50171
32	0.00228	0.05371	0.53997
64	0.00055	0.053713	0.26777
128	0.000286	0.05105	0.26343

From Table 3 it can be appreciated that after executing the GA 50 times, with different number of variables, we can note the best, average and worst results for the F3 function.

Table 3. Experimental results with ga for f3 function

VARIABLES	BEST	AVERAGE	WORST
8	3.006976	3.14677173	3.38354
16	3.163963	3.351902975	3.57399568
32	3.246497	3.14677173	3.86201
64	3.519591	3.86961452	4.15382873
128	3.8601773	4.209902992	4.55839099

From Table 4 it can be appreciated that after executing the GA 50 times, with different number of variables, we can note the best, average and worst results for the F4 function.

Table 4. Experimental results with ga for f4 function

VARIABLES	BEST	AVERAGE	WORST
8	0.499336	6.7430	15.3442
16	8.160601	24.01	43.39
32	46.008504292	82.35724	129.548
64	162.4343	247.0152194	347.216184
128	524.78094	672.6994	890.93943

From Table 5 it can be appreciated that after executing the GA 50 times, with different number of variables, we can note the best, average and worst results for the F5 function.

Table 5. Experimental results with ga for f5 function

VARIABLES	BEST	AVERAGE	WORST
8	0.001547453	0.026897950	0.09962
16	0.0053780643	0.12157227	0.34964841
32	0.141923311	0.410196991	0.9173677
64	0.7874362847	0.980005731	1.00242183
128	1.0051894441	1.006888465	1.00810391

6.2 Simulation Results with Particle Swarm Optimization

The results obtained after applying the PSO to the mathematical functions are shown in tables 6, 7, 8, 9 and 10. From Table 6 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can see the best, average and worst results for the F1 function.

Table 6. Experimental results with PSO for f1 function

VARIABLES	BEST	AVERAGE	WORST
8	1.43E-14	4.21182E-11	9.98E-11
16	6.12E-12	5.31E-11	9.94E-11
32	3.40E-12	5.42E-11	9.86E-11
64	2.01E-12	4.89E-11	9.82E-11
128	3.323E-12	5.34E-11	9.73E-11

From Table 7 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can note the best, average and worst results for the F2 function.

Table 7. Experimental results with PSO for f2 function

VARIABLES	BEST	AVERAGE	WORST
8	2.84E-13	5.20E-11	9.98E-11
16	1.95E-12	4.97E-11	9.96E-11
32	1.93E-12	5.42E-11	9.83E-11
64	5.95E-12	6.12E-11	9.91E-11
128	2.004E-11	8.60E-11	9.55E-11

From Table 8 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can note the best, average and worst results for the F3 function.

Table 8. Experimental results with PSO for f3 function

VARIABLES	BEST	AVERAGE	WORST
8	2.9804998406	3.045430224	3.28918888
16	3.3917876285	3.17059523	3.26680512
32	3.106364703	3.217813872	3.39178762
64	3.227560368	3.379519078	3.55310978
128	3.518976467	3.66857109	3.8473198

From Table 9 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can note the best, average and worst results for the F4 function.

Table 9. Experimental results with PSO for f4 function

VARIABLES	BEST	AVERAGE	WORST
8	7.57E-11	3.661448	8.95462
16	2.984877	11.1837	24.874822
32	16.1450834	34.1697120	56.714207
64	72.364868	126.01692	198.1616
128	368.575586	467.931817	607.874952

From Table 10 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can note the best, average and worst results for the F5 function.

Table 10. Experimental results with PSO for f5 function

VARIABLES	BEST	AVERAGE	WORST
8	4.43E-11	0.00463639	0.04439
16	6.71E-11	0.007300457	0.034526
32	9.17E-06	0.011476878	0.09483
64	0.137781	0.370872190	0.667802
128	0.8560474164	0.970930240	1.003151

6.3 Simulation Results with PPSO

The results obtained after applying the proposed PPSO method to the mathematical functions are shown in Tables 11, 12, 13, 14 and 15.

From Table 11 it can be appreciated that after executing the PPSO 50 times, with different number of variables, we can note the best, average and worst results for the F1 function.

Table 11. Experimental results with PPSO for f1 function

VARIABLES	BEST	AVERAGE				WORST
		CORE 1	CORE 2	CORE 3	CORE 4	
8	4.9738e-014	2.5268e-011	3.4523e-011	2.7674e-011	3.1740e-011	9.0914e-011
16	6.4355e-07	0.003892	0.00446	0.00389	0.00457	0.09853
32	1.7980e-008	0.5141	0.3422	0.3480	0.4207	2.9932
64	0.03182	0.05983	0.084295	0.09507	0.0488	0.6915
128	0.30249	0.5879	0.43788	0.78830	0.893849	0.99456

From Table 12 it can be appreciated that after executing the PPSO 50 times, with different number of variables, we can note the best, average and worst results for the F2 function.

Table 12. Experimental results with PPSO for f2 function

VARIABLES	BEST	AVERAGE				WORST
		CORE 1	CORE 2	CORE 3	CORE 4	
8	9.84E-15	5.20E-11	0.5051	0.0018	0.4852	9.98E-11
16	5.47E-15	7.01E-14	3.50E-13	4.462E-12	8.04E-13	8.66E-10
32	1.45E-13	4.97E-13	2.25E-10	3.78E-10	4.78E-12	6.39E-09
64	2.32E-13	6.28E-11	8.20E-11	5.15E-11	1.28E-11	11.91E-10
128	6.51E-12	3.12E-10	2.31E-10	1.21E-10	6.12E-09	3.55E-07

From Table 13 it can be appreciated that after executing the PSO 50 times, with different number of variables, we can note the best, average and worst results for the F3 function.

Table 13. Experimental results with PPSO for f3 function

VARIABLES	BEST	AVERAGE				WORST
		CORE 1	CORE 2	CORE 3	CORE 4	
8	4.2836e-009	0.01581	0.5051	0.0018	0.4852	1.4555
16	0.0007	0.0494	0.0844	0.0028	0.2589	2.0156
32	0.0513	0.5859	0.0710	1.0122	0.5140	2.5500
64	0.2827	0.3061	3.5063	3.2566	2.5245	3.1528
128	0.3827	0.5061	3.5032	4.2566	6.5245	8.1528

From Table 14 it can be appreciated that after executing the PPSO 50 times, with different number of variables, we can see the best, average and worst results for the F4 function.

Table 14. Experimental results with PPSO for F4 FUNCTION

VARIABLES	BEST	AVERAGE				WORST
		CORE 1	CORE 2	CORE 3	CORE 4	
8	1.28E-15	8.58E-14	1.28E-13	2.36E-13	3.25E-13	1.45E-10
16	5.35E-14	6.25E-13	3.25E-16	3.98E-13	8.85E-13	8.25E-11
32	3.44E-13	1.43E-12	2.54E-12	1.48E-12	1.32E-12	3.64E-10
64	2.12E-07	5.21E-06	1.02E-05	0.02E-05	8.28E-05	3.46E-03
128	1.07E-03	7.77E-02	7.62E-02	1.44E-02	1.25E-02	3.68E-01

From Table 15 it can be appreciated that after executing the PPSO 50 times, with different number of variables, we can note the best, average and worst results for the F5 function.

Table 15. Experimental results with PPSO for F5 FUNCTION

VARIABLES	BEST	AVERAGE				WORST
		CORE 1	CORE 2	CORE 3	CORE 4	
8	4.25E-12	1.44E-11	4.45E-11	2.34E-11	6.30E-11	1.35E-10
16	6.50E-12	2.54E-10	3.43E-10	1.10E-10	3.14E-10	3.75E-09
32	4.25E-12	3.15E-11	5.56E-11	1.52E-11	1.62E-11	7.84E-05
64	6.36E-08	2.52E-07	3.22E-07	3.52E-07	6.20E-07	5.16E-03
128	3.45E-04	6.74E-03	1.48E-03	6.74E-03	1.45E-03	2.39E-01

7 Comparison Results between GA, PSO and PPSO

In Table 16 the comparison of the average results obtained among the GA, PSO and PPSO methods for the optimization of the 5 proposed benchmark mathematical functions is presented. It can be appreciated that the proposed method was better than GA and PSO used separately because when the number of variables for evaluation by each function is large, the adaptation of parameters is necessary. The average results for the functions with 128 variables are shown in Table 16. The best results were obtained with the star topology of PSO.

Table 16. Comparison results between the proposed methods

Function	GA	PSO	PPSO
F1	1.00E-05	3.323E-12	0.30249
F2	0.000286	2.004E-11	6.51E-12
F3	3.860177	3.5118976	0.3827
F4	524.78094	368.57558	1.07E-03
F5	1.0051894	0.8560474	3.45E-04

8 Conclusions

The analysis of the simulation results of the optimization method considered in this paper, PPSO lead us to the conclusion that for the optimization of this benchmark mathematical function with this method is a good alternative because it is easier to optimize achieve good results than to try it with PSO or GA without parameter adaptation. Also, with the configuration of Parallel Particle Swarm Optimization is possible to obtain the results faster and with the PPSO lead us to the conclusion that the best results were obtained with the star topology. Recently we are working with more variables for test the effectiveness of this approach.

Acknowledgment. We would like to express our gratitude to the CONACYT, Institute of Technology for the facilities and resources granted for the development of this research.

References

[1] Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Designs. Springer (1999)

[2] Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995)

[3] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)

[4] Holland, J.H.: Adaptation in natural and artificial system. The University of Michigan Press, Ann Arbor (1975)

[5] Valdez, F., Melin, P.: Parallel Evolutionary Computing using a cluster for Mathematical Function Optimization, Nafips, San Diego CA USA, 598–602 (June 2007)

[6] Castillo, O., Melin, P.: Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. IEEE Transactions on Neural Networks 13(6), 1395–1408 (2002)

[7] Fogel, D.B.: An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks 5(1), 3–14 (1994)

[8] Goldberg, D.: Genetic Algorithms. Addison Wesley (1988)

[9] Emmeche, C.: Garden in the Machine. The Emerging Science of Artificial Life, p. 114. Princeton University Press (1994)

[10] Angeline, P.J.: Using Selection to Improve Particle Swarm Optimization. In: Proceedings 1998 IEEE World Congress on Computational Intelligence, pp. 84–89. IEEE, Anchorage (1998)

[11] Angeline, P.J.: Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 601–610. Springer, Heidelberg (1998)

[12] Back, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Oxford University Press (1997)

[13] Montiel, O., Castillo, O., Melin, P., Rodriguez, A., Sepulveda, R.: Human evolutionary model: A new approach to optimization. Inf. Sci. 177(10), 2075–2098 (2007)

[14] Castillo, O., Valdez, F., Melin, P.: Hierarchical Genetic Algorithms for topology optimization in fuzzy control systems. International Journal of General Systems 36(5), 575–591 (2007)

[15] Valdez, F., Melin, P., Castillo, O.: An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms. Appl. Soft Comput. 11(2), 2625–2632 (2011)

Indirect Adaptive Control with Fuzzy Neural Networks via Kernel Smoothing

Israel Cruz Vega[1], Luis Moreno-Ahedo[1], and Wen Yu Liu[2]

[1] Unidad de Estudios de Posgrado e Investigación
Tecnológico de Estudios Superiores de Coacalco
{iscruz,lmoreno}@ieee.org
[2] Automatic Control Deparment-CINVESTAV
yuw@ctrl.cinvestav.mx

Abstract. In this paper, a neurofuzzy adaptive control framework for discrete-time systems based on kernel smoothing regression is developed. Kernel regression is a nonparametric statistics technique used to determine a regression model where no model assumption has been done. Due to similarity with fuzzy systems, kernel smoothing is used to obtain knowledge about the structure of the fuzzy system and this information is used as initial conditions of the adaptive neurofuzzy control. Results of simulation shows the efficiency of this technique

Keywords: Indirect Adaptive Control, Neural Networks, Fuzzy Systems, Kernel Regression.

1 Introduction

Fuzzy systems and neural networks have been used successfully in nonlinear systems modeling and control [22]. Both are universal approximators that can approximate any continuous function on a compact set to any desired accuracy [23]. Neural networks are used in conjunction with fuzzy systems due to its properties of learning and adaption capabilities [8].

Fuzzy controllers model the control strategy of a human expert to control a system for which no mathematical or physical model exist. They employ a set of linguistic rules to describe the human behavior. The design of fuzzy controllers has no systematic approach [17]. Basically it depends on the available knowledge about the dynamics of the plant to be controlled. One case is if the internal dynamics of the plant (i.e., its differential equations) and their properties are well known, this knowledge can be used to design stable controllers [18] [5]. The advantage is that their stability can be studied and proved, and we obtain a fixed controller. However the design can be a very time-consuming task and the structure needs to be redefined for every different plant, even if they are similar.

Another case, when reduced knowledge of the plant is available (for instance, some properties of the differential equations, such as their upper bounds), alternative methods have to be used, for instance, automatic techniques that are able to obtain the controller parameters and developing algorithms which can learn

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 386–398, 2013.

fuzzy systems automatically from data. The idea of learning has been studied in other research areas like machine learning and data mining; these developed methods are available for the learning process in fuzzy systems. The most important methods are currently derived from statistics [19], cluster analysis [7], or neural network theory [14]. All methods use sample data (data vectors, observation data) to learn from.

In this paper we will develop a new approach in the modeling and control of nonlinear systems with neuro fuzzy systems and kernel smoothing, kernel smoothing is a nonparametric regression technique used in statistics, developed by Nadaraya [12] and Watson [24] with a mathematical fundation given by Parzen's earlier work on kernel density estimation [15]. Kernel regression is the estimation of the functional relationship between two variables y and t on the interval $\{0 \leq t_i \leq T\}, i = 1, ..., N$, given by $y_i = y(t_i) + \xi$, where ξ is a random noise variable with mean equal to zero. Kernel smoothing is used in contribution with an integrated analytic framework for automated fuzzy neural network structure selection and parameter identification, where kernel smoothing provides the base to the structure of the neuro fuzzy system (number of fuzzy sets, bandwidth and center of the fuzzy systems). The structure of the plant is of a particular form $y(k+d) = \alpha(x(k)) + \beta(x(k))u(k)$, where the unknown nonlinear dynamics of the plant $\alpha(x(k))$ and $\beta(x(k))$, will be estimated by kernel regression, and this information will be used as the initial conditions for the adaptive neuro fuzzy control system, which under certain conditions of the plant, the adaptive controller will perform satisfactory.

The sections are organized as follows, section 2 describes the problem, section 3 gives the base for the estimation of the nonlinear functions and generation of fuzzy systems via kernel smoothing, section 4 specifies the design of the indirect adaptive fuzzy controller, section 5 specifies the design of the adaption law, in section 6 is presented the simulations results and in section 7 the conclusions.

2 Problem Description

The plant to be controlled is expressed in terms of its difference equation in a special form,

$$y(k+d) = \alpha(x(k)) + \beta(x(k))u(k) \tag{1}$$

where $\alpha(x(k))$ and $\beta(x(k))$ represent the unknown nonlinear dynamics of the plant, which we assume that are smooth functions. We assume that $\beta(x(k))$ satisfy $0 < \beta_0 \leq \beta(x(k))$ for some $\beta_0 > 0$ for all $x(k)$, this places a restriction on the class of plants that we can consider, where we require that the gain on $u(k)$ be bounded from below due to how an estimate of β will be used to specify the control.

If we assume $r(k+d)$ is known, since this is specified by the user, we can design an ideal controller of the form,

$$u^*(k) = \frac{-\alpha(x(k)) + r(k+d)}{\beta(x(k))} \tag{2}$$

that linearizes the dynamics of the equation (1), such that $y(k) \to r(k)$. Substituying $u(k) = u^*(k)$ in (1) we obtain $y(k+d) = r(k+d)$, so that we achieve tracking of the reference input within d steps. The problem is that $u^*(k)$ is not known because we do not know $\alpha(x(k))$ and $\beta(x(k))$. Here we will use kernel smoothing as an estimator for these plant nonlinearities and use them to form an approximation to $u^*(k)$. In particular, we estimate the unknown $\alpha(x(k))$ and $\beta(x(k))$ mappings with two different linear in the parameter approximators. Let

$$\begin{aligned} \alpha(x(k)) &= \theta_\alpha^{*T}\phi_\alpha(x(k)) + w_\alpha(k+d) \\ \beta(x(k)) &= \theta_\beta^{*T}\phi_\beta(x(k)) + w_\beta(k+d) \end{aligned} \tag{3}$$

where w_α and w_β are the representation errors due to the use of a finite size approximator. Let p be the number of parameters of the approximator, that according to the universal approximator theorem, we will have to experiment to find an approximator simple and flexible enough, to be tuned to find a proper shape of the unknown nonlinearity and make the representation errors arbitrarily small on a closed and bounded set. In our case, the kernel smoothing approximator used will have parameters such as number of fuzzy sets, width and center of the fuzzy sets. We assume that θ_α^* and θ_β^* are the "ideal parameters" for $\alpha(x(k))$ and $\beta(x(k))$, in particular, these are

$$\begin{aligned} \theta_\alpha^* &= \arg\min \theta_\alpha \in \Omega_\alpha \left(\sup_{x \in S_x} \left| \theta_\alpha^T \phi_\alpha(x) - \alpha(x) \right| \right) \\ \theta_\beta^* &= \arg\min \theta_\beta \in \Omega_\beta \left(\sup_{x \in S_x} \left| \theta_\beta^T \phi_\beta(x) - \beta(x) \right| \right) \end{aligned} \tag{4}$$

Here, $\Omega_\alpha \in \Re^{p_\alpha}$ and $\Omega_\beta \in \Re^{p_\beta}$ are the convex and closed bounded sets containing the feasible parameter vectors θ_α and θ_β, respectively. We will use kernel smoothing to estimate the unknown constants θ_α^* and θ_β^* in order to try to specify the controller. $S_x \subset \Re^{n+m+1}$ is assumed to be a closed and bounded set, where we will define our approximator technique to have good accuracy over a range of their domain and let S_x be this region of the domain, where we expect to do good control.

3 Estimating $\alpha(x(k))$ and $\beta(x(k))$ via Kernel Smoothing

Consider a nonlinear discrete plant represented by

$$y(k) = f[\mathbf{x}(k), \theta] + e(k) \tag{5}$$

where $\mathbf{x}(k) = [y(k-1), \cdots, y(k-n_y), u(k-d), \cdots, u(k-d-n_u)]^T \in \Re^{n\times 1}$ and $f(\cdot)$ are unknown nonlinear difference equation representing the plant dynamics, $u(k)$ and $y(k)$ are measurable scalar input and output, respectively, d is the delay, θ is an unknown parameter vector associated with an appropiate but yet to be determine model structure, $e(k)$ is a bounded observation noise, n_y and n_u are the lengths of output and input, $n_y + n_u = n$. In fact, (5) is a NARX model [3].

In the indirect adaptive control case, we consider a special subclass of plants that can be represented by

$$y(k+d) = \alpha(x(k)) + \beta(x(k))\, u(k)$$

where $\alpha(x(k))$ and $\beta(x(k))$ are unknown smooth functions of the state $x(k)$ and represent the unknown nonlinear dynamics of the plant.

A generic fuzzy model is presented as a collection of fuzzy rules in the following form (Mamdani fuzzy model [10])

$$\begin{aligned} R^j &: \text{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots\ x_n \text{ is } A_n^j \\ &\quad \text{THEN } \hat{y} \text{ is } B^j \end{aligned} \tag{6}$$

We use $l\,(j = 1, 2, ..., l)$ fuzzy IF-THEN rules to perform a mapping from the input linguistic vector $x = [x_1, ..., x_n] \in \Re^n$ to the output $\hat{y}$. From [22] we know, by using product inference, center-average and singleton fuzzifier, the output of the fuzzy logic system can be expressed as

$$\hat{y} = \left(\sum_{j=1}^{l} w_j \left[\prod_{i=1}^{n} \mu_{A_i^j} \right] \right) \Big/ \left(\sum_{j=1}^{l} \left[\prod_{i=1}^{n} \mu_{A_i^j} \right] \right) = W\varphi[\mathbf{x}(k)] \tag{7}$$

where $\mu_{A_i^j}$ is the membership functions of the fuzzy sets A_i^j, w_{pj} is the point at which $\mu_{B_{pj}} = 1. W = [w_1, ..., w_l]\,, \varphi$ is $l-$dimension vector function, the element

$$\varphi_i = \prod_{i=1}^{n} \mu_{A_i^j} / \sum_{j=1}^{l} \left(\prod_{i=1}^{n} \mu_{A_i^j} \right) \tag{8}$$

If we use Takagi-Sugeno fuzzy model [1] [21] the fuzzy rules are

$$\begin{aligned} R^j &: \text{ IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots\ x_n \text{ is } A_n^j \\ \text{THEN } \hat{y} &= p_0^j + p_1^j x_1 + \cdots + p_n^j x_n \end{aligned} \tag{9}$$

where $j = 1, ..., l$. The output of the fuzzy logic system can be expressed as

$$\hat{y} = \sum_{i=1}^{l} \left(p_{q0}^i + p_{q1}^i x_1 + \cdots + p_{qn}^i x_n \right) \varphi_i \tag{10}$$

where φ_i is defined as in 8. (10) can be also expressed in the form of the Mamdani-type (7) ,

$$\hat{y} = W\varphi[\mathbf{x}(k)] \tag{11}$$

with $W(k) = \left[p_0^1 \cdots p_n^1\ p_{11}^1 \cdots p_{11}^l\ p_0^l \cdots p_n^l \right]$. Since the Takagi-Sugeno fuzzy model (10) has the same mathematical expression as the Mamdani fuzzy model (7), in this papes we only discuss Mamdani fuzzy system.

Generally the fuzzy networks (7) can not match the given nonlinear system (5) exactly, the nonlinear system (5) can be represented as

$$y\left(k\right)=W\varphi\left[V,\mathbf{x}\left(k\right)\right]+\varepsilon\left(k\right) \tag{12}$$

where $\varepsilon\left(k\right)$ is defined as modelling error. The identified nonlinear system (5) can also be written as

$$y\left(k\right)=W^{0}\varphi\left[V^{0},\mathbf{x}\left(k\right)\right]+\tilde{f}\left(k\right) \tag{13}$$

where $\tilde{f}_k$ is the modelling error, V^0 and W^0 are sets of known parameters chosen by the user. In general, $\left|\tilde{f}\left(k\right)\right|\geq\varepsilon\left(k\right).$ From (13) we know the modelling error de $\tilde{f}_t$ depends on the structure of the fuzzy neural network. Assume we have data of observations $\left[y\left(k\right),\mathbf{x}\left(k\right)\right],$ taken at time points $k\in\left[\tau_1,\tau_2\right].$ We assume that the data come from the model

$$f\left(\mathbf{x}\right)=\phi\left(\mathbf{x}\right)+\epsilon$$

where $\phi\left(\mathbf{x}\right)$ is an unknown smooth mean response curve, and ϵ is the error. Our goal is to find a kernel smooth estimate $\hat{\phi}$ at some pre-specified time point k.

Kernel regression of statistics was derived independently by Nadaraya and Watson [12] with a mathematical foundation given by Parzen's earlier work on kernel density estimation. Kernel regression is the estimation of the functional relationship between two variables y and t on the interval $\left\{0\leq t_i\leq T\right\},i=1,...,N$

$$y_i=y\left(t_i\right)+\xi$$

where ξ is a random noise variable with the mean equal to zero. The Nadaraya-Watson kernel regression estimate at time t is

$$y_i=\frac{\sum_i^N y_i k\left(t-t_i\right)}{\sum_i^N y_i\left(t-t_i\right)},\qquad k\left(t-t_i\right)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{t-t_i}{2\sigma^2}\right] \tag{14}$$

Here the Gaussian density of statistics is the kernel function. For the nonlinear system (5) modeling, (14) is

$$\hat{y}\left(k\right)=\frac{\sum_i^N y_i k\left[\mathbf{x}\left(k\right)-\mathbf{x}\left(t_i\right)\right]}{\sum_i^N k\left[\mathbf{x}\left(k\right)-\mathbf{x}\left(t_i\right)\right]}$$
$$k\left[\mathbf{x}\left(k\right)-\mathbf{x}\left(t_i\right)\right]=\frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{\left\|\mathbf{x}\left(k\right)-\mathbf{x}\left(t_i\right)\right\|^2}{2\sigma^2}\right] \tag{15}$$

where $\hat{y}\left(k\right)$ is the estimated value of $y\left(k\right)=f\left[\mathbf{x}\left(k\right)\right].$

For the data set $\left[y\left(t_i\right),\mathbf{x}\left(t_i\right)\right]$, we extract the rules in the form of (6). $\mathbf{x}\left(t_i\right)$ is the center of the gaussian function $\mu_{A_i^j}$, $y\left(t_i\right)$ is the center of the gaussian function μ_{B^j}. If we compare (15) with (7), kernel smoothing is equivalent to fuzzy modeling if each data corresponds to one fuzzy rule. However, this kind of fuzzy structure cannot be applied, because there are too many fuzzy rules from kernel smoothing when n is large. A posible way is design a wide kernel function to cover more data, i.e., groups of data, such that one group of data correspond to one rule.

A practical method is *full width at half maximum.* The maximum width is defined as $\sigma_{\max} = 2\sqrt{2\ln(2)}\sigma$, $\sigma_{\max}$ is chosen by the total number of data, for example $\sigma_{\max} = \frac{N}{10}$, the width parameter of the gaussian function is $\sigma = \frac{N}{20\sqrt{2\ln(2)}}$.

We use center points of each Gaussian function $\left[x_j^*, y_j^*\right], j = 1, ..., v$ to construct the membership functions. For the data set $\left[y(k), \mathbf{x}(k)\right], k \in [1, N]$, we extract fuzzy product rules in the form of (6). x_j^* is the center of the gaussian function $\mu_{A_i^j}$, y_j^* is the center of the gaussian function μ_{B^j}.

4 Indirect Adaptive Fuzzy Controller

The usefulness of fuzzy systems for control has been proven in the past. Nevertheless, there still exists the drawback of having to define the system parameters specifically for each plant.

The universal-approximation property of fuzzy systems implies that it is possible to achieve any desired approximation accuracy by properly adding more membership functions and rules. With this idea we can obtain controllers capable of learning the control policy from the beginning, with a fast convergence.

Based on the use of online approximation with kernel regression we propose an indirect adaptive control scheme. We will focus on developing the controller to track the reference input $r(k)$ where $|r(k)|$ is bounded by a finite constant and the tracking error is defined as $e(k) = r(k) - y(k)$. The control scheme is shown in figure 1.

Considering the discrete-time system of the form 1, we propose the following controller using an estimate of the plant model that would cancel appropiate plant dynamics and achieve good tracking with an accurate estimate,

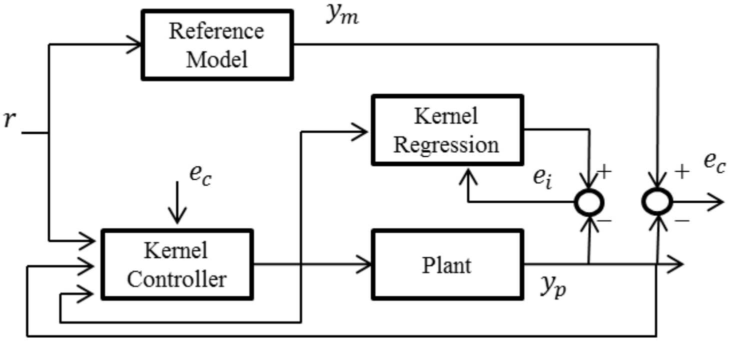


Fig. 1. Indirect Adaptive Control

$$u(k) = \frac{-\hat{\alpha}(x(k)) + r(k+d)}{\hat{\beta}(x(k))} \tag{16}$$

where $\hat{\alpha}(x(k))$ and $\hat{\beta}(x(k))$ are estimates of $\alpha(x(k))$ and $\beta(x(k))$, respectively, defined as

$$\begin{aligned} \hat{\alpha}(x(k)) &= \theta_\alpha^T(k)\,\phi_\alpha(x(k)) \\ \hat{\beta}(x(k)) &= \theta_\beta^T(k)\,\phi_\beta(x(k)) \end{aligned} \tag{17}$$

we will use gradient optimization methods to pick $\theta_\alpha(k)$ and $\theta_\beta(k)$ to try to minimize the approximation error.

Now, we derive an expression for the tracking error $e(k) = r(k) - y(k)$ and if we advance time by d steps

$$\begin{aligned} e(k+d) &= r(k+d) - y(k+d) \\ &= r(k+d) - \alpha(x(k)) - \beta(x(k))\,u(k) \\ &= (\hat{\alpha}(x(k)) - \alpha(x(k))) + \\ &\quad \left(\hat{\beta}(x(k)) - \beta(x(k))\right) u(k) \end{aligned}$$

where we used the value of $r(k+d)$ obtained from equation (16). Now, if we define

$$\hat{y}(k+d) = \hat{\alpha}(x(k)) + \hat{\beta}(x(k))\,u(k)$$

then using this equivalent control law (16), the output tracking error becomes

$$e(k+d) = \hat{y}(k+d) - y(k+d)$$

which can be view as the "identification error", i.e., it is a measure of the quality of the model that we are tunning to represent the plant.

The parameter errors for the indirect adaptive controller are defined as

$$\begin{aligned} \tilde{\theta}_\alpha(k) &= \theta_\alpha(k) - \theta_\alpha^* \\ \tilde{\theta}_\beta(k) &= \theta_\beta(k) - \theta_\beta^* \end{aligned}$$

then, with these definition, the tracking error becomes

$$\begin{aligned} e(k+d) = {} & \tilde{\theta}_\alpha^T(k)\,\phi_\alpha(x(k)) + \tilde{\theta}_\beta(k)\,\phi_\beta(x(k))\,u(k) \\ & -w_\alpha(k+d) - w_\beta(k+d)\,u(k) \end{aligned} \tag{18}$$

Here we can see that the representation errors affect the tracking error. $e(k+d)$ is measurable, since it is assume that $y(k+d)$ and $r(k+d)$ are measurable, but $\tilde{\theta}_\alpha(k)$ and $\tilde{\theta}_\beta(k)$ are not known because we assume that we do not know the ideal parameters.

Now, in order to find a linear error equation that the optimization method seeks to minimize, first consider

$$\tilde{\theta} = \left[\tilde{\theta}_\alpha^T, \tilde{\theta}_\beta^T\right]^T$$

and

$$\theta(k) = \left[\theta_\alpha^T(k), \theta_\beta^T(k)\right]^T$$

and

$$\phi(x(k), u(k)) = \left[\phi_\alpha^T(x(k)), \phi_\beta^T(x(k))\,u(k)\right]^T$$

where $\phi(x(k), u(k))$ is a function of $u(k)$ due to how we multiply $u(k)$ by $\phi_\beta^T(x(k))$, with this me ensure that we get a linear equation below.

Now, let $w(k) = w_\alpha(k) + w_\beta(k)\,u(k-d)$, where this equation represents the "combined" representation error that arises from approximating both $\alpha(x(k))$ and $\beta(x(k))$.

With this, equation (18) becomes a linear error equation (with an offset defined by the representation error)

$$e(k+d) = \tilde{\theta}^T(k)\,\phi(x(k), u(k)) - w(k+d) \tag{19}$$

with this type of linear error equation we can apply gradient and recursive least squares methods and be assured we can get stable adaptive systems.

5 Design of Adaption Law

Now we will use the normalized gradient method for the training of $\theta(k)$ to try to approximate $\alpha(x(k))$ and $\beta(x(k))$. This is a steepest descent approach with a special choice for the step size. Consider the cost function

$$J(\theta) = \frac{1}{2}e^2(k)$$

which is a measure of the size of the tracking error (and identification error) that we want to minimize, then the update formula woud be $\theta(k) = \theta(k-d) + \lambda_k d(k)$, where λ_k is the step size and $d(k)$ is the descent direction (d here is a constant scalar delay, and $d(k)$ is the notation for a $p \times 1$ vector descent direction). Note that the delay d appears in $\theta(k-d)$ due to the fact that $e(k)$ depends on $\theta(k-d)$. Now, choosing a normalized gradient approach and hence,

$$\begin{aligned} d(k) &= -\frac{\partial J(\theta)}{\partial \theta} \\ &= -e(k)\frac{\partial e(k)}{\partial \theta} \\ &= -e(k)\frac{\partial}{\partial \theta}\left(\theta^T(k-d)\,\phi(x(k-d), u(k-d))\right) \\ &= -e(k)\,\phi(x(k-d), u(k-d)) \end{aligned}$$

Now, we pick λ_k as the selection of the step size with a normalized gradient approach [11], use the notation $\phi(k-d) = \phi(x(k-d), u(k-d))$, and use a "robustification" approach to modify the tracking error and produce $e_\epsilon(k)$, the adaption law becomes

$$\theta(k) = \theta(k-d) + \frac{\kappa_1 \eta \phi(k-d)}{1 + \gamma\left|\phi(k-d)\right|^2} e_\epsilon(k) \tag{20}$$

where $\kappa_1 = -1, \gamma$ is a design parameter, and we assume that the constant "adaption gain" η is such that $0 < \eta < 2\gamma$ this constraint on the adaption gain is specified to help ensure stability. You should think of it as a step size parameter where we need $\eta > 0$ to ensure that we use the negative gradient and we need $\eta < 2\gamma$ to ensure that the step size is small enough to maintain stability.

The Kernel Indirect Adaptive Control System is shown in the figure 2.

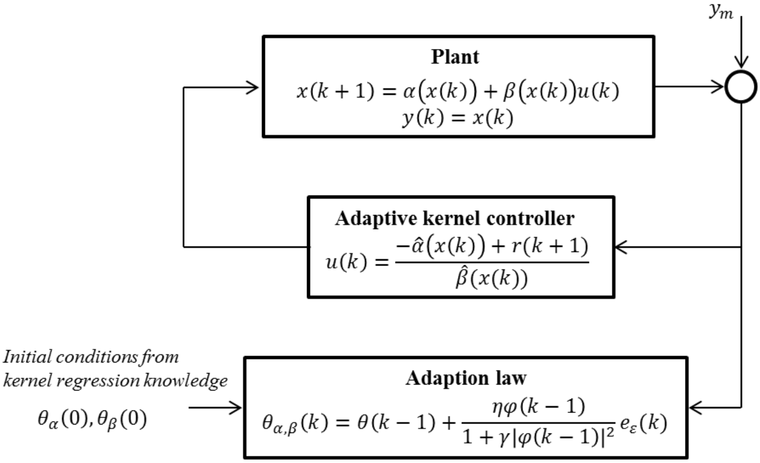

y_m
Plant
$x(k+1) = \alpha(x(k)) + \beta(x(k))u(k)$
$y(k) = x(k)$
Adaptive kernel controller
$u(k) = \frac{-\hat{\alpha}(x(k)) + r(k+1)}{\hat{\beta}(x(k))}$
Initial conditions from kernel regression knowledge
$\theta_\alpha(0), \theta_\beta(0)$
Adaption law
$\theta_{\alpha,\beta}(k) = \theta(k-1) + \frac{\eta\varphi(k-1)}{1+\gamma|\varphi(k-1)|^2} e_\varepsilon(k)$

Fig. 2. Kernel Indirect Adaptive Control

6 Simulation

The gas furnance data set from Box-Jenkins [2] has been used extensively as benchmark example for process identification. The data set consists of 296 input-output measurements sampled at a fixed interval of 9 seconds. The measured input $u(k)$ represents the flow rate of the methane gas in a gas furnance and the output measurement $y(k)$ represents the concentration of the carbon dioxide in the gas mixture flowing out of the furnance under a steady air supply. The continuous data were collected to provide information about the dynamics of the system over a region of interest where it was known that an approximately linear steady-state relationship applied.

In order to represent the gas furnance data in the form of (1), which is a special class of linear discrete-time systems with constant but unknown coefficients. In particular, let $d = 1$ and

$$\begin{aligned} y(k+1) = {} & a_1 y(k) + a_2 y(k-1) + \cdots + a_{n+1} y(k-n) \\ & + b_1 u(k) + b_2 u(k-1) + \cdots + b_{m+1} u(k-m) \end{aligned}$$

where the $a_i, i = 1, 2, ..., n+1$, and $b_i, i = 1, 2, ..., m+1$, are unknown but constant scalars. If we let

$$\alpha(x(k)) = a_1 y(k) + a_2 y(k-1) + \cdots + a_{n+1} y(k-n) + \\ b_2 u(k-1) + \cdots + b_{m+1} u(k-m)$$

and

$$\beta(x(k)) = b_1$$

then

$$y(k+1) = \alpha(x(k)) + \beta(x(k)) u(k)$$

so that the model clearly fits the form of equation (1). Once we have express the gas furnance data in the form of (1), we will use kernel regression in order to approximate the unknown functions $\alpha(\cdot)$ and $\beta(\cdot)$. For the kernel regressor, the bandwidth of the gaussian functions is 0.5, which produces the less mean squared error (MSE). The results of the identification process are shown in the figure 3.

The fuzzy rules generated by kernel regression are:

$$R_\alpha : \text{IF } x \text{ is } A_\alpha \text{ THEN } \alpha \text{ is } B_\alpha$$
$$R_\beta : \text{IF } x \text{ is } A_\beta \text{ THEN } \beta \text{ is } B_\beta$$

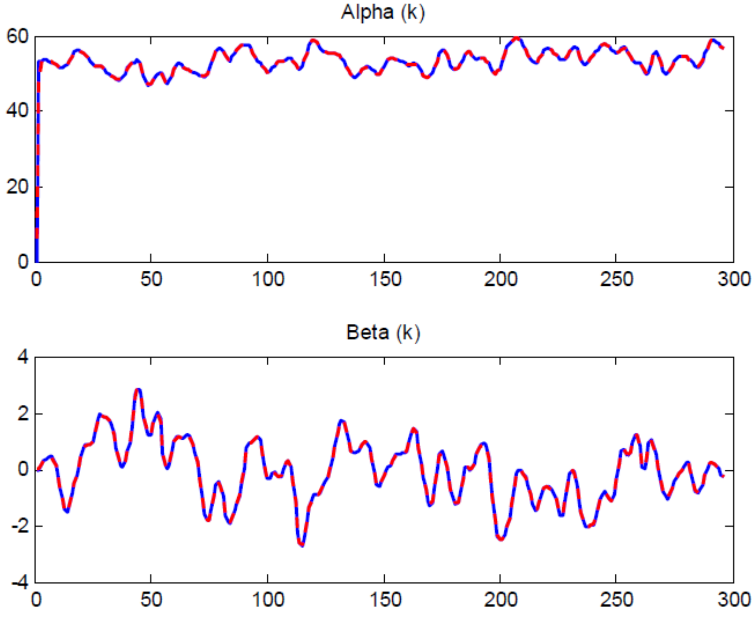


Fig. 3. Kernel identification of nonlinear functions

which generate the initial conditions:

$$\hat{\alpha}_0 = \frac{\sum_{i=1}^{m} b_\alpha \prod_{j=1}^{n} \mu_{A_\alpha}}{\sum_{i=1}^{m} \prod_{j=1}^{n} \mu_{A_\alpha}}; \hat{\beta}_0 = \frac{\sum_{i=1}^{m} b_\beta \prod_{j=1}^{n} \mu_{A_\beta}}{\sum_{i=1}^{m} \prod_{j=1}^{n} \mu_{A_\beta}}$$

In our previous work [4], is compared the performance of our algorithm (SFNN) approach with others identification methods like normal neural networks identification (MLP) [13], multiple RBF neural networks (MNN) [6] [9] and multi-stage neural identification (MSNN) [20].

The multilayer neural network as in [13] is $\Pi_{5,15,1}$ (one hidden layer with 15 nodes), and its learning rate of backpropagation is fixed as $\eta = 0.05$. For multiple neural networks [9], the two neural networks are chosen as $\Pi_{5,15,1}$ and $\Pi_{5,10,1}$, their learning rates are 0.05. The results are shown in Table 1, where the modeling error is calculated as RMS.

The controller was designed using the estimate of the unknown functions by kernel regression that cancel the appropiate plant dynamics and achieve good tracking of the reference signal $r(k)$ with an accurate estimate. The form of the controller is given in the equation (16). The input reference $r(k)$ is a square signal with magnitude of 5. The output of the plant with the reference is shown in the figure 4.

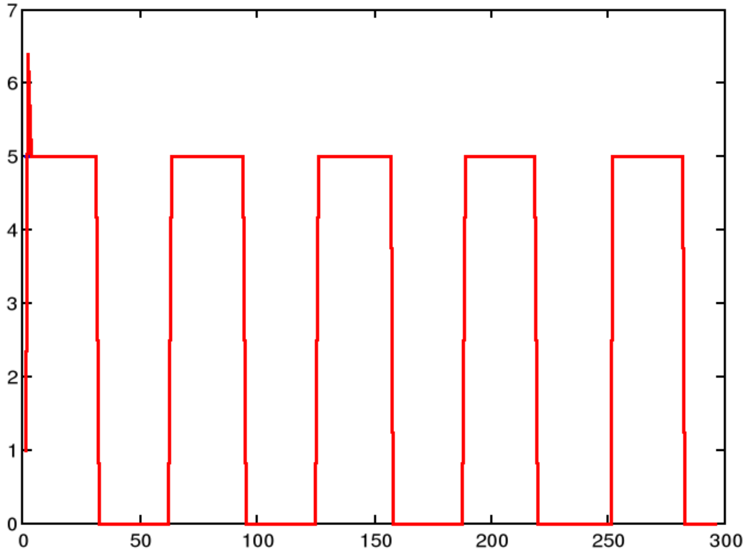


Fig. 4. Plant's output and reference signal

Table 1. Comparisons of four identification methods

Networks	MLP		MNN		MSNN		SFNN	
Case	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Modeling error	0.34	0.62	0.23	0.59	0.08	0.53	0.26	0.48
Learning rate	0.05	–	0.05	–	–	–	1.0	–
Running steps	5000	1000	5000	1000	1000	1000	1000	1000
Hidden nodes Rule number	15	15	10, 15	15, 10	6	6	4, 6	4, 6

7 Conclusions

In this work we have used kernel regression as an easy tool for regression and control of nonlinear systems. Because of the similarity of kernel regression with fuzzy regression, kernel is used to obtain structural information of fuzzy systems in size, position and number of membership functions. Based on the use of online approximation with kernel regression we propose an indirect adaptive control scheme, where kernel regression is used in the approximation of the nonlinear functions which will be used in the control law in order make the systems follow the reference signal. Results show that kernel regression is easy to implement and in the knowledge of the structure of the neuro-fuzzy system. Kernel regression technique is only used for initial conditions of the adaptive neuro fuzzy controller, which will perform better under change conditions of the plant.

References

1. Baruch, I.S., Lopez, R.B., Olivares, J.-L., Flores, J.M.: A fuzzy-neural multi-model for nonlinear systems identification and control. Fuzzy Sets and System 159, 2650–2667 (2008)
2. Box, G.E.P., Jenkins, J.M., Reinsel, G.C.: Time series analysis, forecasting and control. Prentice-Hall, Inc., Upper Saddle River (1994)
3. Chen, S., Billings, S., Grant, P.: Recursive hybrid algorithm for nonlinear system identification using radial basis function networks. Int. J. Control 55, 1051–1070 (1992)
4. Cruz, I., Yu, W., Cordova, J.J.: Multiple fuzzy neural networks modeling with sparse data. In: 2010 IEEE International Conference on Fuzzy Systems (FUZZ), July 18-23 (2010)
5. Galluzo, M., et al.: Control of biodegradation of wixed wastes in a continuous bioreactor by a type-2 fuzzy logic controller. Comput. Chem. Eng. 33(9), 1475–1483 (2009)

6. Haykin, S.: Neural Networks, A Comprehensive Foundations. Prentice Hall, Englewood Cliffs (1999)
7. Höppner, F., Klawonn, F., Kruse, R.: Fuzzy-Clusteranalyse, Computational Intelligence. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig (1996)
8. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall (1997)
9. Kikens, B., Karim, M.: Process identification with multiple neural network models. Int. J. Control 72, 576–590 (1999)
10. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. IEEE Proccedings-Control Theory and Applications 121, 1585–1588 (1976)
11. Mandic, D.P., Hanna, A.I., Razaz, M.: A Normalized Gradient Descent Algorithm for Nonlinear Adaptive Filters Using a Gradient Adaptive Step Size. IEEE Signal Processing Letters 8(11) (2001)
12. Nadaraya, E.A.: On estimating regression. Theory of Probability and its Applications 9, 141–142 (1964)
13. Narendra, K.S., Cheng, X.: Adaptive control of discrete-time systems using multiple models. IEEE Transactions on Automatic Control 45(9), 1669–1686 (2000)
14. Nauck, D., Klawonn, F., Kruse, R.: Foundations of Neuro-Fuzzy Systems. John Wiley & Sons, Inc., New York (1997)
15. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Statistics 33, 1065–1076 (1962)
16. Priestley, M.B., Chao, M.T.: Non-parametric function fitting. J. Royal Statistical Soc., Ser. B 34, 385–392 (1972)
17. Rajapakse, A., Furuta, K., Kondo, S.: Evolutionary learning of fuzzy logic controllers and their adaption through perpetual evolution. IEEE Trans. Fuzzy Sys. 10(3), 309–321 (2002)
18. Lalouni, S., Rekioua, D., et al.: Fuzzy logic control of stand-alone photovoltaic system with battery storage. J. Power Sources 193(2), 899–907 (2009)
19. Werner, L.: Einführung in die Regelungstechnik. Friedr. Vieweg & Sohn VerlagsgesellschaftmbH, Braunschweig (1992)
20. Li, K., Peng, J., Bai, E.W.: A two-stage algorithm for identification of nonlinear dynamic systems. Automatica 42(7), 1189–1197 (2006)
21. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst., Man, Cybern. 15, 116–132 (1985)
22. Wang, L.X.: Adaptive Fuzzy Systems and Control. Prentice-Hall, Englewood Cliffs (1994)
23. Wang, L.X.: A Course in Fuzzy Systems and Control, 2nd edn. Prentice Hall, Upper Saddle River (1997)
24. Watson, G.S.: Smooth Regression Analysis. Sankhya Ser. A. 26, 101–116 (1964)

Search and Detection of Failed Components in Repairable Complex Systems under Imperfect Inspections

Boris Kriheli and Eugene Levner

Askelon Academic College, Ashkelon, Israel
borisk@hit.ac.il, elevner@acad.ash-college.ac.il

Abstract. We study a problem of scheduling search-and-detection activities in complex technological and organizational systems. Prior probabilities of failures are known for each element, and the decision maker has to sequentially inspect the components so that to find a failed component within a given level of confidence. The inspections are imperfect; namely, a probability of overlooking the failed element and a probability of a "false alarm" exist. An index-based algorithm for finding the optimal search strategy is developed. An example for robotic search systems is discussed.

Keywords: Search-and-detection, Discrete search, Imperfect inspections, False-positive detection, False-negative detection, Greedy algorithm.

1 Introduction

The need to search for hidden or lost objects arises in many civilian and military applications. Suppose a fault occurs in a complex technological system containing several components. Until it is eliminated, the fault causes damage to the whole system, the damage scale being dependent on the location of the failed component and the time needed to detect the failed component. The problem is to efficiently detect and remove or repair the failed component.

A similar situation is often encountered in practice in many areas other than the fault diagnostics. Rescue teams may search for lost persons and shipwrecks. Autonomous agents like mobile robots should find victims and survivors after natural disasters. In military search operations, mobile agents, like patrol aircraft and unmanned aerial vehicles (UAVs), look for high-valued targets such as lost submarines, missile launchers or terrorist locations. Similar problems arise in planning risky R&D projects, medical diagnostics, and search for natural resources.

In all of these applications, the choice of a search strategy strongly influences the search costs and losses incurred by the damage/harm as well as the time for finding the target within a specific time limit. In military search missions with UAVs, the problem of finding good strategies for UAVs performing the search-and-detection mission is especially complicated because, like manned aircraft, these mobile agents need to operate in hostile environments. Under hostile, sabotage and insidious

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 399–410, 2013.

circumstances, two types of errors in search-and-detection tests usually occur: (i) a so-called "false-negative" detection wherein a target object or location is overlooked; and (ii) a "false-positive detection" (also called "a false alarm") which wrongly classifies a good object as failed or malicious. Unfortunately, the problem of selecting the "best" search strategy is fundamentally hard due to its stochastic nature and the nonlinearity induced by the detection probability. In particular, looking twice into the same location by a searcher generally does not double the detection probability. Recent reviews and texts provide a deeper background and further applications of search problems (see, e.g., Washburn [22], Kress et al. [12], Lee [15], and Chung and Burdick [5]).

In this work, we study a stochastic search-and-detection process subject to the false-negative and false-positive inspection outcomes. We are interested in organizing the search process by using locally optimal, or greedy, strategies called index-based strategies [1, 11]. This is a sequential strategy wherein, at each step, the decision maker computes a current "search effectiveness" for each component and recommends searching next a component with the highest current effectiveness. It is attractive because of its simplicity and computational efficiency; moreover, such local search strategy guarantees finding an optimal search sequence with a given confidence level for the problem considered in this paper. The meaning of the latter terms is explained in Section 3.

The remainder of the paper is organized as follows. In the next section, we provide a review of the discrete sequential search problems and their recent re-emergence in autonomous mobile-agent and robotic systems. Then we develop a necessary theoretical basis and present problem formulation in Section 3. In Section 4, we propose a mathematical model and a greedy strategy that partly incorporates the information obtained from observations, while optimizing the search cost without significant computational load. A small numerical example related to the robotic search is presented in Section 5. A summary of contributions and an outline of future research are provided in Section 6.

2 Related Work

The discrete search problem is one of the oldest problems of Operations Research and Artificial Intelligence. Its initial study was made over seven decades ago by Bernard Koopman and his team during the World War II aimed to provide efficient methods for detecting submarines [10]. The search theory has been used by the US Navy in the search for the H-bomb lost in the ocean near Palomares, Spain, in 1966. A historical survey of classical search theory is given in the seminal works of Washburn [22] and Stone [20]. Recent research includes various search problems with novel civil and military applications. We refer to [2, 5-7, 11-15, 19] for recent studies of discrete search algorithms and complexity results.

Much of the early research relies on the assumption that only false-negative detections ("the overlooks") occurs while the false-alarm probability is zero. Kadane [8] and Stone [20] considered a situation in which times and probabilities of fault detections change during the search. Kadane [8] examined the optimal “whereabouts” search problem of locating a stationary target or, if unsuccessful within a fixed search budget, specifying its most likely location. Chew [3], Wegener [23], and Levner [16] also only considered false-negative detection probabilities but with varying costs that are associated with inspecting different components.

Chew [3] and Kadane [8] considered a situation where the probability of detection is to be maximized subject to a cost constraints and showed that a locally optimal strategy is optimal when the cost is measured in the number of searches. Kadane and Simon [9] proposed a unified approach to the min-cost and max-reward (finite-horizon) search problems.

Trummel and Weisinger [21] and Wegener [24] showed that the general minimum-expected-time-to-detection and cost-with-switches search problems are NP-hard. The criteria for the termination of the search process, called the optimal stopping rule, were treated by Chew [4] and Ross [18]; however, such termination conditions still rely on the assumption of zero false-positive detections and, thus, only correctly capture the false-negative search process.

The robotics and military OR communities have also made great strides in the problem of sequential search. Research presented in [2, 5, 6] and the references therein presented a Bayesian construction for the problem of searching for multiple lost targets by robots, using the probability of detection of these targets as the objective for optimal search trajectory. However, this work also does not address the presence of false-positive detections, assuming that any positive detection always reveals a true target.

In recent years, the studies of discrete search problems were focused on the design of new mathematical models for non-standard military applications. Kress et al. [11] considered a team of searchers comprising two groups: searchers that find threat targets and disposable UAVs or ground units used to destroy the targets when they are found. Kress et al. [13] and Lee [15] considered a group of sensors that searches for stationary threat objects such as ballistic missile launchers or improvised explosive devices. The task of the sensor is to determine if an area cell contains a threat object, and the objective of the searcher is to maximize the number of correctly determined area cells subject to false-positive and false-negative test results. Eagle and Yee [6], and Sato and Royset [19] formulated and solved discrete-time path-optimization problems where a searcher, operating in a discrete three-dimensional airspace, looks for a moving target in a finite set of cells. The searcher is constrained by limits on the consumption of resources such as time, fuel, and risk along any path. These authors developed branch-and-bound algorithms that use network reductions and the Lagrangean relaxation.

Kress et al. [11] and Chung and Burdick [5] have studied in-depth the false-positive detections in discrete-search problems. The specific contributions of [11] are the following: (a) the authors show that a local index-based policy is optimal when each positive detection by the (autonomous) search sensor is followed by an investigation by a human verification team; (b) if the verification time is prohibitively longer than the search time, an alternative measure of effectiveness, namely, the probability that the first positive detection is a true one, is introduced and analyzed. However, these authors neither treat the false-negative error type nor study a stopping rule in the case where the human verification team is absent. In contrast with this work, in what follows, we study a more general situation wherein the two types of errors can occur and no human inspection team is being involved.

A main contribution of [5] is the decision-theoretic representation of the spatio-temporal search process, where the sequentially gathered observations drive the searcher's search decision in the search area. The evolution of the search decision provides a unified approach for both the affirmative search problems, i.e., finding an target known to be present in an area, as well as the search quitting option, specifying a termination criterion to stop the search process and declare the object absent. Another major contribution of [5] is the evaluation of several proposed (myopic and biology-inspired) search strategies minimizing the search time. Complementary to this research, in the present paper we will consider a more general search cost objective function, develop a greedy index-based strategy for its minimization and prove its optimality. The main contribution of the present paper is that its mathematical model is quite general and the algorithm developed in the paper includes, as special cases, a number of earlier algorithms developed for perfect inspections ([8, 17]), as well as for the imperfect inspections with only false-negative outcomes ([1, 3, 9, 16, 17. 20, 22]) and only false-positive outcomes ([11]).

3 Problem Formulation

In this paper, the sequential discrete search problem is defined as follows. A discrete search area of interest (AOI) may be a complex technological system or a set of possible locations of a stationary and distinguishable target. The subject of the search is a failed or lost or hidden component (target). One and only one target is to be found within the considered framework. The target location in the search area is uncertain, namely, there are nonzero probabilities that the target is located in given locations; they are available to the decision maker before the search commences. Given the stationary nature of the target, the object is assumed to be present throughout the entire duration of the search process, i.e., the object does not leave the search area.

We assume that that an inspection of any (either failed or good) element is imperfect. This means that there given a prior probability α_i of the false alarm and, in addition, there given a prior probability β_i of overlooking the failed element. This implies that examination of each element can happen more than once. In other words, a search sequence, in general, may be infinite. When the failure occurs, a single mobile searcher should perform a set of sequential inspections in order to identify the failed component. The times and costs of inspections of all elements being given, the *goal of the search* is to determine a search strategy which the searcher should employ in order to locate the target (failed element) with the minimum or near-minimum cost.

Let us give a formal description of the problem.

A system contains N modules $1,2,\ldots N$. The input data are the following: When the system fails, each element i, $i = 1,\ldots, N$, is characterized by the following known parameters:

p_i - prior probability that element i is failed;

α_i - prior probability of a "false alarm" , or a false-positive outcome, that is, a conditional probability that an inspector (it can be a sensor, a robot, an UAV, etc.) declares that element i is a target (for instance, failed) whereas in fact it is not a target (not failed);

β_i - prior probability of overlooking, or a false-negative outcome, that is a conditional probability that an inspector declares that element i is not a target whereas in fact it is the target;

t_i – expected time to inspect element i;

c_i - search cost rate per unit time when searching element i.

Each sequential inspection strategy specifies an infinite sequence

$$s = <S[0], s[1], s[2], \ldots, s[n], \ldots>,$$

where $s[n]$ denotes the element (more exactly, the number of the element) which is inspected at the n-th step of sequence s, all $s[n] \in \{1,2,\ldots, N\}$; $S[0]$ denotes an initial sub-sequence of elements which will be defined below.

Given the above input data, the optimal search scenario is specified by the following conditions:

(i) the elements are inspected sequentially;

(ii) for any search strategy and any failure location, the outcomes of inspections are independent;

(iii) *the stopping rule* is defined as follows:

(a) First, for any integer h, we define the conditional probability $a_{[i,h]}$ that element i is really failed under condition that the inspector declares that the i is failed in h inspections; the $a_{[i,h]}$ depends on given p_i, α_i and β_i and is computed in the next section;

(b) Second, a *confidence level CL* is given *a priori*, say CL =0.95, and for each element, parameter H_i called a "height" is defined as the minimum positive integer such that the probability $a_{[i,h]}$ exceeds the confidence level CL, that is, $a_{[i,h]} \geq CL$. (Notice that all H_i can be computed by the decision maker before the search process starts);

(c) In a sequence s, the search ends when, at some step, for some element i, the outcome of inspection is: "the element i is failed for the H_i-th time in s".

For a given sequence s, we shall use the following notation:

$T_{s[n]} = T(s[n],s)$ – (accumulated) time spent to detect the failed component $s[n]$ on the n-th step of strategy s; $T_{s[n]} = \sum_{m=1}^{n} t_{s[m]}$.

$P_{\underline{s}[n]}$ – probability that an element $s[n]$ is detected as failed $H_{s[n]}$ times up to the n-th step of strategy s. (The values $H_{s[n]}$ and $P_{s[n]}$ depend on parameters α_i and β_i, and guarantee a required confidence level; this property will be explained in detail below);

$F(s)$ - the expected total search cost.

In accordance with the above conditions (i) and (ii) of the considered search scenario, the expected (linear) total search cost, $F(s)$, is defined as follows:

$$F(s) = \sum_{n=1}^{\infty} P_{s[n]} c_{s[n]} T_{s[n]}.$$

In the above notation, the stochastic scheduling problem studied in this paper is to find a sequence s^* minimizing the expected total search cost $F(s)$. In what follows, the terms *strategy s* and *sequence s* will be used interchangeably.

When all the $\alpha_i = \beta_i = 0$ we have a special case of so-called perfect inspections extensively studied in scheduling literature (see, for example, Kadane and Simon [9], and Rabinowitz and Emmons [17].). When all $\alpha_i = 0$, but $\beta_i \neq 0$, this is a case of false negative inspections; if $_{\alpha i}\neq 0$, but all $\beta_i = 0$, we have only false-positive inspections.

4 Problem Analysis and Algorithm

We start with introducing the following notation:

Event B_i ={Inspector declares that element i is failed}.

Event C_i ={Element i is really failed}.

Using the aforementioned notation of Section 3, probability of C_i and probabilities of the errors of two types are expressed as follows:

$$p_i = P(C_i);\ \alpha_i = P\left(B_i / \overline{C}_i\right);\ \beta_i = P\left(\overline{B}_i / C_i\right).$$

The probability that element *i* is detected as failed is equal to

$$f_i = P(B_i) = P(C_i)P(B_i / C_i) + P(\overline{C}_i)P(B_i / \overline{C}_i) =$$
$$= p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i$$

The probability to correctly detect the failed element *i* within a single inspection is equal to

$$P(C_i / B_i) = \frac{P(C_i)P(B_i / C_i)}{P(C_i)P(B_i / C_i) + P(\overline{C}_i)P(B_i / \overline{C}_i)} =$$
$$= \frac{p_i \cdot (1-\beta_i)}{p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i}$$

The following claim is straightforward:

Claim 1. Given a sequence *s,* the probability $a_{[i,h]}$ that a module *i* is really failed under condition that it is discovered to be failed within exactly *h* inspections in *s* (where *h* is some integer) satisfies the following relations:

$$a_{[i,h]} = P\left(C_i / B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{(h)}\right) =$$
$$= \frac{P(C_i) \cdot P\left(B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{(h)} / C_i\right)}{P\left(B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{h}\right)} =$$
$$= \frac{P(C_i)P^h\left(B_i^{(1)} / C_i\right)}{P(C_i)P^h\left(B_i^{(1)} / C_i\right) + P(C_i)P^h\left(B_i^{(1)} / \overline{C}_i\right)} = \frac{p_i \cdot (1-\beta_i)^h}{p_i \cdot (1-\beta_i)^h + (1-p_i)\alpha_i^h}$$

Corollary. Given a predetermined confidence level *CL* (say, *CL* =0.95), the 'height' H_i for any element i is computed as the minimum integer satisfying the following condition:

$$a_{[ih]} = \frac{p_i \cdot (1-\beta_i)^{H_i}}{p_i \cdot (1-\beta_i)^{H_i} + (1-p_i)\alpha_i^{H_i}} \geq CL$$

The search strategy is defined as an (infinite) sequence of module numbers, where, at each step *n*, a module *s*[*n*] is inspected and tested whether or not is failed;

$$s = <S[0], s[1], \ldots, s[n], \ldots>.$$

In this sequence, *S*[0] denotes an initial sub-sequence defined in such a way that the probability to stop during this sub-sequence is zero. For instance, *S*(0) can be selected as follows:

$$S[0] = \left\{ \underbrace{1,1,,,1,}_{H_1-1}\ \underbrace{2,2,\ldots 2,}_{H_2-1} \ldots,\ \underbrace{N, N, \ldots N}_{H_N-1} \right\}$$

Claim 2. Any permutation of elements 1,2,…*N* in the above sub-sequence S[0] preserves the needed property that the probability to stop during a (new) sub-sequence is zero.

The proof is evident and skipped here.

Now we are capable to compute the conditional probability $P_{s[n]}$ that an element *s*[*n*] is detected as failed exactly H_i times at the *n*-th step of strategy *s* if *s*[*n*] is known to be failed. (Recall that this probability is needed for computing the problem's objective function *F*(*s*)). For this aim, we need to introduce an auxiliary parameter *s**[*n*]. For any given *s* and *s*[*n*], let *s**[*n*] be the total number of inspections of module *s*[*n*], up to the *n*-th step of strategy *s*; evidently, $s^*[n] \leq n$, for all *n*. Notice that *s**[*n*] can be easily computed as soon as the sequence *s* is known up to the *n*-th step.

It is evident that

$$\begin{cases} s^*[n] \geq H_{s[n]} & when \qquad n \geq 1 \\ s^*[0] < H_{s[0]} & \end{cases}$$

Claim 3. Probability $P_{s[n]}$ that an element *s*[*n*] is detected as failed $H_{s[n]}$ times up to the *n*-th step of strategy s is determined as follows:

$$P_{s[n]} = \begin{cases} \binom{s^*[n]-1}{H_{s[n]}-1} (1-f_{s[n]})^{s^*[n]-H_{s[n]}} (f_{s[n]})^{H_{s[n]}}, n \geq 1 \\ 0 \qquad , \qquad n = 0 \end{cases}$$

This claim immediately follows from the above definitions, and the binomial distribution of $H_{s[n]}$-1 inspections of the element $s(n)$, with the outcome "failed", within a total number $s^*[n]$ of inspections of the element $s[n]$ up to the n-th step of s.

Now we can define more exactly all the components of the problem's objective function $F(s)$.

The time $T_{\mathrm{s[n]}}$ spent for the inspection of all the elements $S[0],\ldots, s[n]$ up to the n-th step of strategy s is the following:

$$T_{s[n]} = \sum_{i=1}^{N} t_i (H_i - 1) + \sum_{m=1}^{n-1} T_{[m,s]} + t_{s[n]} \qquad n \ge 1$$

The probability $P_{s[n]}$ that an element $s[n]$ is detected as failed exactly H_i times at the n-th step of strategy s is defined in Claim 3.

The search cost attributed to strategy s is

$$F(s) = \sum_{n=1}^{\infty} c_{s[n]} \cdot T_{s[n]} \cdot P(element\ s[n] is \det ected\ as\ failed\ H_{s[n]}\ times) =$$

$$= \sum_{n=1}^{\infty} c_{s[n]} \cdot T_{s[n]} \cdot P_{s[n]} =$$

$$= \sum_{n=1}^{\infty} c_{s[n]} \cdot T_{s[n]} \cdot \binom{s^*[n]-1}{H_{s[n]}-1} \cdot (1 - f_{s[n]})^{s^*[n]-H_{s[n]}} (f_{s[n]})^{H_{s[n]}}.$$

For the validity of the algorithm below we will need the following property.

Claim 4. Let h be an integer and

$$P_{f_i}(h) = \binom{h-1}{H_i - 1} (1 - f_i)^{h-H_i} (f_{s[n]})^{H_i},\ h \ge H_i.$$

$P_{f_i}(h)$ is a non-increasing function of h, for $h \ge H_i$, if and only if $H_i f_i \le 1$.

The proof of Claim 4 is achieved by elementary algebraic transformations.

The main contribution of this paper is the greedy algorithm that permits to find the optimal strategy according to the following claim.

Theorem 1. Let $H_i f_i \le 1$ for all components i. The strategy s^* is an optimal strategy iff the ratios

$$Q_{s[n]} = \frac{c_{s[n]}}{t_{s[n]}} \cdot \binom{s^*[n]-1}{H_{s[n]}-1} (1 - f_{s[n]})^{s^*[n]-H_{s[n]}} (f_{s[n]})^{H_{s[n]}}$$

are arranged in non–increasing order, for all $n \ge 1$.

Proof. 1. Assume that $s[n] \neq s[n+1]$ for some n in sequence s^* specified by Theorem 1. Consider two adjacent strategies, s_1 and s_2, where $s_1 = s^*$, and s_2 is obtained from s_1 by swapping elements $s[n]$ and $s[n+1]$:

$$s_1 = <S[0], s[1], \ldots., s[n\text{-}1], s[n], \mathrm{s}[n+1], s[n+2],\ldots>,$$

$$s_2 = <S[0], s[1], \ldots., s[n\text{-}1], s[n+1], s[n], s[n+2],\ldots>$$

To prove the theorem it is sufficient to prove the following statement:

$$F(\mathrm{s}_1) \leq F(\mathrm{s}_2) \Leftrightarrow Q_{\mathrm{s}[n]} \geq Q_{\mathrm{s}[n+1]}.$$

We have:

$$F(s_1) - F(s_2) =$$

$$-c_{s[n]} \cdot t_{s[n+1]} \cdot \binom{s^*[n]-1}{H_{s[n]}-1} \cdot (1-f_{s[n]})^{s^*[n]-H_{s[n]}} \left(f_{s[n]}\right)^{H_{s[n]}} +$$

$$c_{s[n+1]} \cdot t_{s[n]} \cdot \binom{s^*[n+1]-1}{H_{s[n+1]}-1} \cdot (1-f_{s[n+1]})^{s^*[n+1]-H_{s[n+1]}} \left(f_{s[n+1]}\right)^{H_{s[n+1]}} \leq 0 \Leftrightarrow$$

$$c_{s[n]} \cdot \binom{s^*[n]-1}{H_{s[n]}-1} \cdot (1-f_{s[n]})^{s^*[n]-H_{s[n]}} \left(f_{s[n]}\right)^{H_{s[n]}} / t_{s[n]} \geq$$

$$c_{s[n+1]} \cdot \binom{s^*[n+1]-1}{H_{s[n+1]}-1} \cdot (1-f_{s[n+1]})^{s^*[n+1]-H_{s[n+1]}} \left(f_{s[n+1]}\right)^{H_{s[n+1]}} / t_{s[n+1]}$$

It follows that $F(s_1) - F(s_2) \leq 0 \Leftrightarrow Q_{s[n]} \geq Q_{s[n+1]}$. We proved that $F(s_1) \leq F(s_2) \Leftrightarrow Q_{s[n]} \geq Q_{s[n+1]}$. This claim is true for all $n \geq 1$.

2. Assume now that for some $n \geq 1$ in strategy s^* specified by Theorem 1 there exists an element i^* such that $i^* = s[n] = s[n+1]$, and, at the same time, the values $Q_{s[n]}$ and $Q_{s[n+1]}$ defined by the theorem are different. Then, using elementary algebraic transformations, we obtain that if condition $H_i f_i \leq 1$ holds then $Q_{s[n]} \geq Q_{s[n+1]}$. It follows that Theorem 1 assigns a correct order of $s[n]$ and $s[n+1]$, namely, element i^* will be selected in s^* at the $(n+1)$-th step if and only if it has been earlier selected at the n-th step. The optimality of the index-based strategy is proved. □

Corollary. 1) The problem of minimizing the search cost $F(s)$ is well defined since $F(s^*)$ is bounded from above when $n \to \infty$.

2) The search process defined by Theorem 1 quickly converges in the sense that the probability of the event that the search does not stop at step $s[n]$ exponentially decreases as the number n grows.

Notice that we call the search policy in Theorem 1 greedy or index-based, because, in order to find an optimal sequence s^*, at each step, the decision maker selects for inspecting the next element with the maximal ratio.

The algorithm presented in Theorem 1 generalizes and includes, as special cases, several earlier models and algorithms developed by different authors for the case of perfect inspections ([9,17]), as well as for the imperfect inspections with only false-negative outcomes ([1, 3, 7, 8, 16, 20]), and only false-positive outcomes ([11]).

5 Example: Robotic Search for a Hidden Target

Consider a problem of searching for a target by a single robot in a stochastic setting described in [2, 5]. An autonomous mobile device, e.g., an UAV, is able to function without the direct intervention of human beings and has control over its own actions. The specificity of the considered robotic search is that this autonomous device has a limited memory and does not remember all the outcomes of its previous search steps. It only saves in memory how many times a target has been detected in each visited location up to a current step in the search sequence. The search stops as soon as the number of such outcomes in any one of the locations reaches its pre-specified limit, H_i. Unnecessary data transmission between the robot and the control center can be avoided since the optimal search strategy s^* is known to the robot or can be easily computed on board. Even if the connection with the control center fails, the mobile agent can successfully follow its optimal sequence and perform its mission.

An area of interest is divided into N possible locations in one of which a target object is hidden. Further details of the search for the hidden target by an autonomous device are given in [2, 5]. For simplicity, we consider an area with only three locations numbered 1, 2 and 3. The input data are given in Table 1; a confidence level $CL = 0.95$; for simplicity, all c_i=1. The problem is to find the optimal search strategy.

Table 1. Input data

	Locations		
	1	2	3
$p_i = P(C_i$	0.1	0.15	0.75
$\alpha_i = P(B_i / \overline{C}_i);$	0.04	0.06	0.12
$\beta_i = P(\overline{B}_i / C_i)$	0.1	0.07	0.05
t_i	5	8	10

The results of computations are given in Table 2.

Table 2. Numerical results

	Locations		
	1	2	3
H_i	2	2	1
$f_i =$ $p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i$	0.126	0.190	0.743

Using the aforementioned formulas of Section 3 we obtain that $S[0] = <1,2>$, $s^*[1] =1$, $s^*[2]=2$, $s^*[3]=1$, $s^*[4]=2$; $s^*[5] = s^*[6] = s^*[7] = s^*[8] = 3$, $s^*[9] = 2,\ldots$; $Q_{s[1]} = 0.1527$, $Q_{s[2]} = 0.0819$, $Q_{s[3]} = 0.0384$, $Q_{s[4]} = 0.0312$, $Q_{s[5]} = 0.0257$; $Q_{s[6]} = 0.0191$, $Q_{s[7]} = 0.0141$, $Q_{s[8]} = 0.0105$, $Q_{s[9]} = 0.0089$, … .Thus, the optimal strategy derived up to the ninth step is the following: $s^* = <1,2,1,2,1,2,3,3,3,2,3\ldots>$.

This example shows that the ratios Q_i (as well as the probability that the search stops in an incorrect location caused by false alarms) quickly decrease as the number of steps n grows. Notice that the searcher does not use in this setting a full history of all outcomes obtained at each step of the search sequence. Incorporating this information into the search model leads to a more complex dynamic programming setting that falls out of the scope of this paper and will be explored in future research.

6 Conclusions and Future Work

We study a stochastic search-and-detection process for hidden or failed objects subject to false-negative and false-positive test outcomes. For optimizing the search process, we use a greedy strategy called the index-based strategy which is proven to be optimal when the objective is to minimize the expected cost of the search-and-detection. This is a sequential strategy which, at each step, computes a current "search effectiveness" for each component and recommends to inspect next the component with the highest current effectiveness. Being attractive because of its computational efficiency and simplicity, such local search strategy guarantees finding an optimal (minimum-cost) search sequence with an arbitrary pre-specified confidence level.

A main contribution of this paper is three-fold. First, we develop a mathematical model of the discrete-time search process under two types of uncertainties, false-positive and false-negative ones. Second, we formulate a new stopping rule according to which the search process terminates when, at a certain step, for some element i, the sensor reports that the outcome of inspection is "failed" sufficiently many times (H_i times) guaranteeing a predetermined confidence level. Third, a new fast algorithm for finding the optimal strategy is developed.

This work gives a starting point for exploring other search scenarios and models. A potential future research is to compare the efficiency of different search methods using analytical and simulation tools. Different search scenarios (e.g., with multiple mobile agents, multiple targets, moving targets, resource constraints, precedence relations, agents-with-memory, etc) are of theoretical and practical importance. Advanced solution methods, such as dynamic programming, branch-and-bound, biology-motivated algorithms, and stochastic programming, can be employed for solving these problem types.

References

1. Assaf, D., Zamir, S.: Optimal sequential search: A Bayesian approach. Ann. Statist. 13(3), 1213–1221 (1985)
2. Bourgault, F., Furukawa, T., Durrant-Whyte, H.F.: Optimal Search for a Lost Target in a Bayesian World. In: Yuta, S., et al. (eds.) Field and Service Robotics. STAR, vol. 24, pp. 209–222. Springer, Heidelberg (2006)

3. Chew Jr., M.C.: A sequential search procedure. Annals of Mathematical Statistics 38(2), 94–502 (1967)
4. Chew Jr., M.C.: Optimal stopping in a discrete search problem. Operations Research 21(3), 741–747 (1973)
5. Chung, T.H., Burdick, J.W.: Analysis of search decision making using probabilistic search strategies. IEEE Transactions on Robotics 28(1), 1–13 (2012)
6. Eagle, J.N., Yee, J.R.: An optimal branch-and-bound procedure for the constrained path, moving target search problem. Operations Research 38(1), 110–114 (1990)
7. Ferrari, S., Fierro, R., Perteet, B., Cai, C., Baumgartner, K.: A geometric optimization approach to detecting and intercepting dynamic targets using a mobile sensor network. SIAM Journal of Control and Optimization 48(1), 292–320 (2009)
8. Kadane, J.B.: Optimal whereabouts search. Oper. Res. 19(4), 894–904 (1971)
9. Kadane, J.B., Simon, H.A.: Optimal strategies for a class of constrained sequential problems. The Annals of Statistics 5, 237–255 (1977)
10. Koopman, B.: The theory of search, II.Target detection. Oper. Res. 4, 324–346 (1956)
11. Kress, M., Lin, K.Y., Szechtman, R.: Optimal discrete search with imperfect specificity. Mathematical Methods of Operations Research 68(3), 539–549 (2008)
12. Kress, M., Royset, J.O., Rozen, N.: The eye and the fist: Optimizing search and interdiction. European Journal of Operational Research 220(2), 550–558 (2012)
13. Kress, M., Szechtman, R., Jones, J.S.: Efficient employment of non-reactive sensors. Military Operations Research 13(4), 45–57 (2008)
14. Lau, H., Huang, S., Dissanayake, G.: Discounted mean bound for the optimal searcher path problem with non-uniform travel times. European Journal of Operational Research 190(2), 383–397 (2008)
15. Lee, K.K.: Efficient employment of adaptive sensor. Monterey, Calif. Naval Postgraduate School (2008)
16. Levner, E.: Infinite-horizon scheduling algorithms for optimal search for hidden objects. Interanational Transactions in Operational Research 1(2), 241–250 (1994)
17. Rabinowitz, G., Emmons, H.: Optimal and heuristic inspection schedules for multistage production systems. IIE Transactions 29(12), 1063–1071 (1997)
18. Ross, S.M.: A problem in optimal search and stop. Oper. Res. 17(6), 984–992 (1969)
19. Sato, H., Royset, J.O.: Path optimization for the resource-constrained searcher. Naval Research Logistics 57, 422–440 (2010)
20. Stone, L.D.: Theory of Optimal Search. ORSA, New York (1992)
21. Trummel, K.E., Weisinger, J.R.: The complexity of the optimal searcher path problem. Operations Research 34(2), 324–327 (1986)
22. Washburn, A.R.: Search and Detection, 4th edn. Topics in Operations Research Series. INFORMS (2002)
23. Wegener, I.: The discrete sequential search problem with nonrandom cost and overlook probabilities. Mathematics of Operations Research 5(3), 373–380 (1980)
24. Wegener, I.: Optimal search with positive switch cost is NP-hard. Information Processing Letters 21(1), 49–52 (1985)
25. Wilson, K.E., Szechtman, R., Atkinson, M.P.: A sequential perspective on searching for static targets. European Journal of Operational Research 215(1), 218–226 (2011)

Distance Aproximator Using IEEE 802.11 Received Signal Strength and Fuzzy Logic

Carlos Fco Álvarez Salgado, Luis E. Palafox Maestre, Leocundo Aguilar Noriega, and Juan R. Castro

Universidad Autónoma de Baja California
Calzada Tecnológico 14418, Tijuana, México 22300
{fco_alvarez,lepalafox,laguilar,jrcastro}@uabc.edu.mx

Abstract. Wireless localization systems based on IEEE 802.11 are becoming more and more common in recent years, due in part to low costs in hardware and effortlessness of deployment with off the shelve Access Points (AP), such localization systems are based on Received Signal Strength (RSS) using a periodic beacon containing information about the source where a signal strength value can be obtained upon reception of this beacon; shadow attenuation effect and multipath fading influences RSS when indoors becoming a random variable dependent on the location of the antennas with a distinguishing statistical distribution called Rayleigh distribution; this article takes upon the measurement process of the distance from AP to a device, where soon after this a position could be resolved by triangulation or trilateration on the device by means of more AP's. This paper proposes a method that uses fuzzy logic for modeling and dealing with noisy and uncertain measurements.

Keywords: Fuzzy Logic, IEEE 802.11, Distance Aproximator.

1 Introduction

Wireless localization systems based on IEEE 802.11 are becoming more and more common in recent years[1], due in part to low costs in hardware and effortlessness of deployment with off the shelve Access Points (AP) and no need for alteration of the hardware or software, such localization systems are based on Received Signal Strength (RSS) [2][3]using a periodic beacon containing information about the source where a signal strength value can be obtained upon reception of this beacon, since receivers are only listening no extra traffic to the network is send, an already deployed Wi-Fi network could participate as a reference point with just knowing the position of the AP; shadow attenuation effect and multipath fading influences RSS when indoors becoming a random variable dependent on the location of the antennas with a distinguishing statistical distribution called Rayleigh distribution; this article takes upon the measurement process of the distance from AP to a device using this characteristic distribution of the signal to estimate the distance, where soon after this a position could be resolved by triangulation or trilateration [4][5]on the device by means of more AP's, the process of resolving the position in not part of this paper. This paper

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 411–420, 2013.

proposes a method that uses a Takagi-Sugeno-Kang (TSK) Fuzzy Inference Systems (FIS) type model for dealing with noisy and uncertain measurements.

Section 2 reviews concepts in Wi-Fi positioning and effects in the received signal strength; methodology is explained in sections 3 and section 4 presents the results and conclusion of the paper.

2 Background

2.1 Wi-Fi Positioning

Wi-Fi (IEEE 802.11) Positioning is an alternative technology to GPS, using radio signals provided by networks found in the urban zone in order to evaluate a position[1][6]. The beacons of private or public AP's are used as reference. Private and public AP's are increasing in the cities; expanding the Wi-Fi coverage every year, providing more accuracy by means of more reference points.

This type of positioning works inside buildings; the precision depends of the number of reference AP's, achieving up to 2 meters, having some issues for the deployment;

Issues with the deployment are:

- Radio coverage and strength could be affected by walls, motors o other electromagnetic interference
- The location of the AP is not known
- The position could change in time
- Some techniques required a training phase.

Based on the method for calculating the position two main types are defined in [7], Fingerprinting or cartography used in [8] and wave propagation models as in [9], being trilateration where a distance measure is needed to evaluate the position the more common one.

Fingerprinting needs a database of measurements of all the mapped area, and access to it to evaluate the position. While trilateration needs the location of three AP's to evaluate positioning.

2.2 Received Signal Strength

Every AP is emitting a beacon to inform potential clients the information needed for connection such as AP name, network type, speeds allowed, security aspects among other information, the AP transmit the beacon with an occurrence of 10 to 20 times a second depending the AP model, when the receiver capture this beacon, the network card register the signal strength of the receiving package.

The Received Signal Strength is measure in dbm meaning decibels relative to 1 mW, the dbm increment is in decibels (db) in a logarithmic manner.

$$RSS_{dbm} = 10log_{10}P \tag{1}$$

Where RSS_{dbm} is the end point Receive Signal Strength in dbm and P is the power received in mW, 1 mW is 0 dbm, signals weaker than 1 mW are negative values.

2.3 Rayleigh Fading and Shadowing

Rayleigh fading[10] is caused by multiple paths, scattering and reflecting of the signal[8]. When no signal path dominates a radio wave cancelation of the same signal exist, the received signal strength becomes a random variable, this fading will vary according with a Rayleigh distribution.

2.4 Log-Normal-Shadow Model

A commonly used model is the log-Normal-Shadow model used in [11][12][13][14]. In this model the RSS $P_r(d)$expressed in dbm at a given distance d from the transmitter is given by:

$$P_r(d) = P_t[dbm] - \overline{PL}(d_0) - 10nlog_{10}\left(\frac{d}{d_0}\right) - X_\sigma \qquad (2)$$

Where $P_t$is the power transmitted signal, $\overline{PL}(d_0)$ is the measured path-loss at reference point d_0,n is the path-loss exponent, and X_σ is a normal random variable with mean 0 and σ variance, this last variable models the path-loss variation across the distance d due to shadowing.

Some variants of this model have being published such as Okumura-Hata model [9] among others.

3 Methodology

Data acquisition intended for comparison and modeling were taken in two scenarios, one a hostile environment location with several APs at various distances from the reference AP in a line-of-sight (LOS) with no obstacles, and an open site with minimal interference for the second location scenario. A Linksys Wireless-G 2.4 GHz is used as a source using 802.11g as a default setting, the receiving beacons were captured by an Acer Travelmate 4650 running Ubuntu Linux; the packages were capture using wireshark software; activating the wireless network card in promiscuous mode capturing all beacons received by the network card.

More than 45,000 individual measurements where capture at distances from 1 to 20 meters, at each meter mark an average of 1,200 measurements were taken from the receiving device of the reference AP, no other filter was used.

A relationship is observed between distance and RSS as shown in fig. 1, measurements near the source have a stronger RSS; having a logarithmic declination as the distance is increased. This logarithmic behavior is used by log-normal-shadow model.

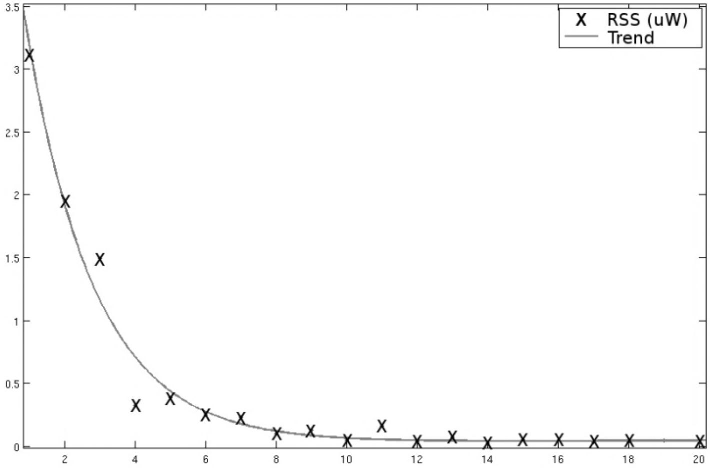


Fig. 1. Mean of measures RSS

3.1 RSS Distribution

Due to environment variables (Rician fading or Rayleigh fading) the value of RSS will fluctuate while in a stationary position fig. 2. Over time when a sufficient amount of packages are received, a left skewed probabilistic distribution will be form, this function is called Rayleigh distribution.

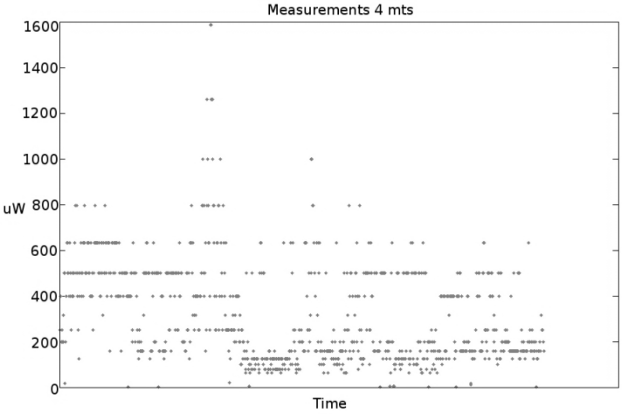


Fig. 2. Measurements scattering at 4 Meters

The skew of the Rayleigh distribution vary meter by meter[8], this is shown in the fig. 3. When the PDF is analyzed a wider spread distribution appears at distances near the source, going narrower as farther it goes; the difference between mean and mode also change, being noticeable greater at short distances while reducing the difference as distance grow displayed in fig. 4. These two differences could be used to estimate the distance between source and receiver.

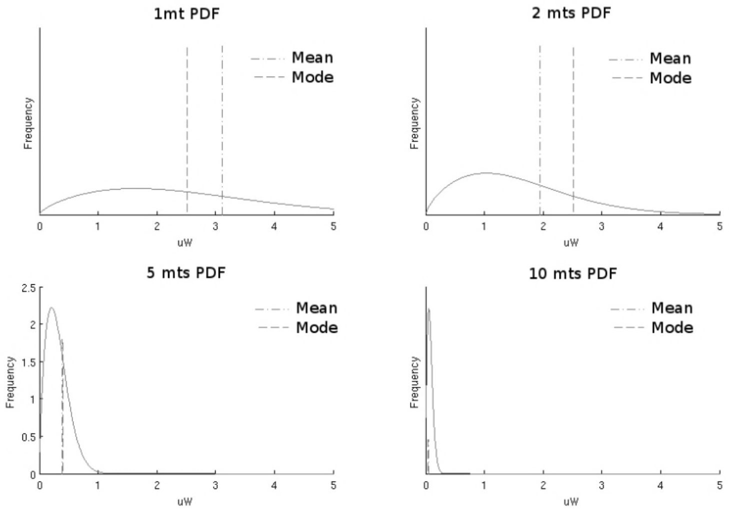


Fig. 3. RSS PDF at 1,2,5 and 10 Meters

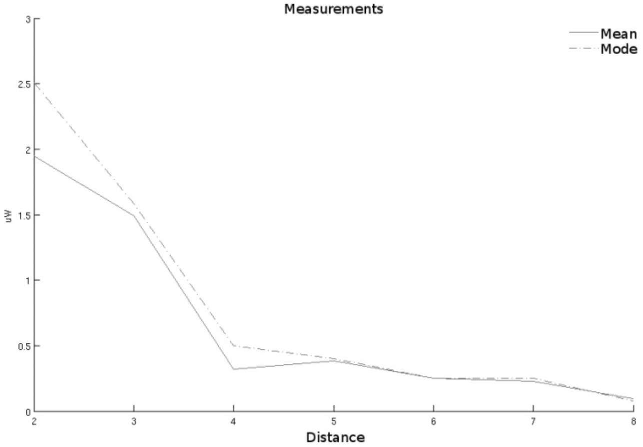


Fig. 4. RSS Mean and Mode over Distance

3.2 Generation of Fuzzy Rule Set

Once the inputs for a Takagi-Sugeno-Kang (TSK) Fuzzy Inference Systems (FIS) type model are selected, these will be used by the FIS to tackle the uncertainty in measurements.

The TSK model is formed with sets of fuzzy rules and functional type consequents decomposing the input space in subspaces and approximate the system in a simple linear regression model, providing an efficient model for complex systems.

For a TSK FIS type model express a logical association between inputs and output thru the set of rules with antecedents and consequents which are used to produce the desired output. To produce this output the antecedent value is a variable of the consequent.

The TSK model is composed of set of IF-THEN rules in the form:

$$R_r\text{: } if\ x_1\ is\ C_r^1\ and\ if\ x_2\ is\ C_r^2\ and\ \ldots and\ if\ x_m is\ C_r^m\ then\ y_r\ is\ f_r(x) \tag{3}$$

Where

$$fr(x) = \alpha_r^0 + \alpha_r^1 x_1 + \cdots + \alpha_r^m x_m \tag{4}$$

In which r is the rule number (r= 0,1, …,n) and x_j $(1 \leq j \leq m)$ are the input variables, y_r is the output variable, $C_r^m$are the fuzzy sets, and $f_r(x)$ is a linear function.

With this equation the TSK model describe each rule as a local lineal behavior associated with inputs as antecedents.

For an input $\hat{x} = \{x_r^1, x_r^2, \ldots, x_r^m\}$the inferred value of the TSK is calculated:

$$\gamma = \frac{\sum_{r=1}^{m} c_r(\hat{x})^*}{\sum_{r=1}^{m} c_r(\hat{x})} = \frac{\sum_{r=1}^{m} \tau_r * f_r(\hat{x})}{\sum_{r=1}^{m} \tau_r} \tag{5}$$

Where τ_r being the firing level of the rth rule for the input$\hat{x}$; Integrating the inference and defuzzycation in a single step process. The TSK rule interpretation depends on the choice of centers and standard deviations of the memberships where the inputs are projected in an axis orthogonal manner.

To divide de input space in clusters, each one representing a subspace of the system, a clustering method is used against the measured data. When projecting the cluster in to the input space the antecedent part of the fuzzy rule could be found. The consequent is a linear can be a simple function so one cluster correspond to one rule of the TSK model.

Subtractive clustering method is used to group the input data space and create the center and sigma of each group cluster, centers and sigma of each cluster form the membership function for the rules to be used by TSK model.

Using the Subtractive clustering with the obtain data the result is a 16 clusters in 16 rules TSK model shown in fig. 5 and fig. 6 obtaining a correlation of 0.94488.

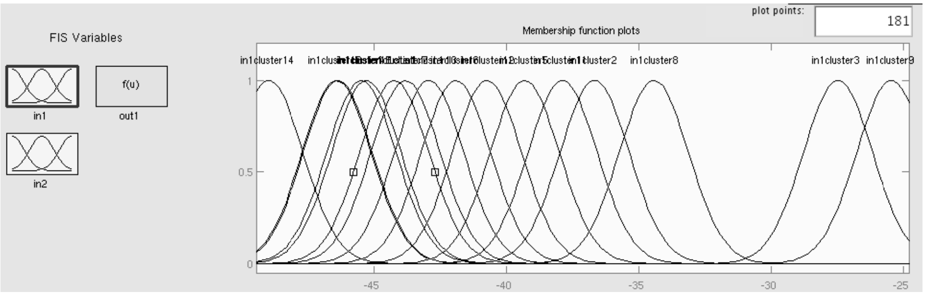
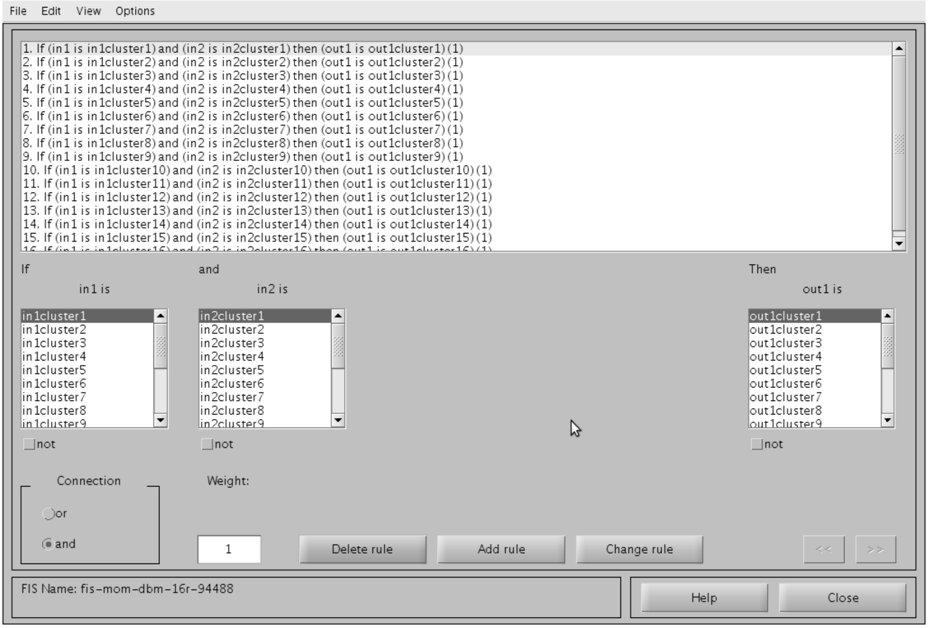


Fig. 5. TSK input clusters



1. If (in1 is in1cluster1) and (in2 is in2cluster1) then (out1 is out1cluster1) (1)
2. If (in1 is in1cluster2) and (in2 is in2cluster2) then (out1 is out1cluster2) (1)
3. If (in1 is in1cluster3) and (in2 is in2cluster3) then (out1 is out1cluster3) (1)
4. If (in1 is in1cluster4) and (in2 is in2cluster4) then (out1 is out1cluster4) (1)
5. If (in1 is in1cluster5) and (in2 is in2cluster5) then (out1 is out1cluster5) (1)
6. If (in1 is in1cluster6) and (in2 is in2cluster6) then (out1 is out1cluster6) (1)
7. If (in1 is in1cluster7) and (in2 is in2cluster7) then (out1 is out1cluster7) (1)
8. If (in1 is in1cluster8) and (in2 is in2cluster8) then (out1 is out1cluster8) (1)
9. If (in1 is in1cluster9) and (in2 is in2cluster9) then (out1 is out1cluster9) (1)
10. If (in1 is in1cluster10) and (in2 is in2cluster10) then (out1 is out1cluster10) (1)
11. If (in1 is in1cluster11) and (in2 is in2cluster11) then (out1 is out1cluster11) (1)
12. If (in1 is in1cluster12) and (in2 is in2cluster12) then (out1 is out1cluster12) (1)
13. If (in1 is in1cluster13) and (in2 is in2cluster13) then (out1 is out1cluster13) (1)
14. If (in1 is in1cluster14) and (in2 is in2cluster14) then (out1 is out1cluster14) (1)
15. If (in1 is in1cluster15) and (in2 is in2cluster15) then (out1 is out1cluster15) (1)
16. If (in1 is in1cluster16) and (in2 is in2cluster16) then (out1 is out1cluster16) (1)

If in1 is: in1cluster1, in1cluster2, in1cluster3, in1cluster4, in1cluster5, in1cluster6, in1cluster7, in1cluster8, in1cluster9 — □ not

and in2 is: in2cluster1, in2cluster2, in2cluster3, in2cluster4, in2cluster5, in2cluster6, in2cluster7, in2cluster8, in2cluster9 — □ not

Then out1 is: out1cluster1, out1cluster2, out1cluster3, out1cluster4, out1cluster5, out1cluster6, out1cluster7, out1cluster8, out1cluster9 — □ not

Connection: ○ or ● and

Weight: 1

FIS Name: fis-mom-dbm-16r-94488

Fig. 6. Rule base TSK model

3.3 Results

A second set of measurements were introduced to test the model. For comparison a reference model was needed. The selected model for comparison was the popular used log-normal-shadow model, a commonly used model [11][12][13][14] with a Mean Square Error (MSE) of 3 meters.

From the set of measurements a mean and mode values were calculated, used as inputs to the TSK FIS model; the mean was the only input to the Log-Normal-

Shadow. The collected results of the MSE for the Log-Normal-Shadows is 2.90 meters, as expected from documented research on the model, the TSK model MSE is a 1.88 meters, a considerable reduction in the MSE.

The linear correlation coefficient R for TSK model is .94488 and .86859 for the log-normal shadow model.

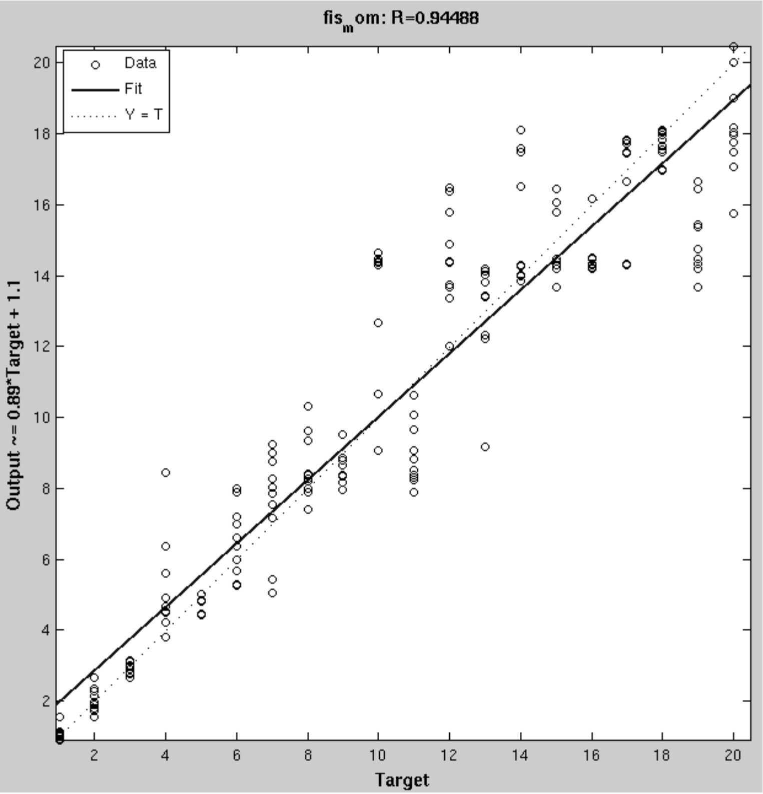


Fig. 7. Plot regression TSK FIS model

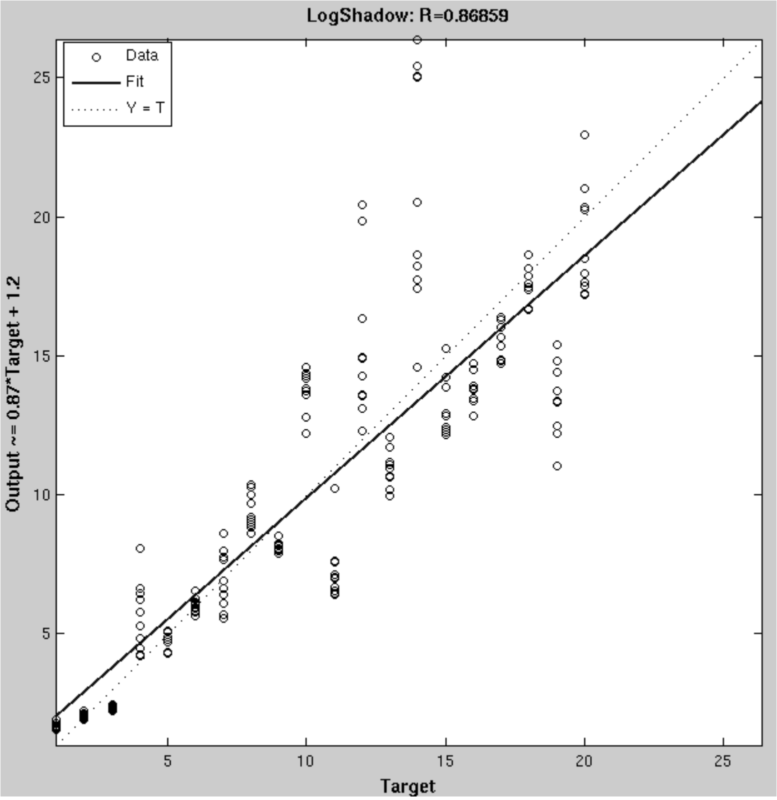


Fig. 8. Plot regression for Log-Normal Shadow Model

3.4 Conclusions

This paper focus on the measuring distance section of positioning, presenting an option for measuring distances between two IEEE 802.11 devices. A future work is the implementation of this model to a complete positioning algorithm to compare results against known methods of measurement like in [15][16].

The TSK model worked in this paper reduce the MSE of a commonly used model, narrowing the error for use in other techniques for trilateration positioning systems based on RSS, helping in accuracy by reducing error margin.

The TSK model takes the uncertainty of the measurements at a computational cost, a deepest study of the use and computational cost of the model is needed for future work, the use of ANFIS could offer a better form of adaptation to a set of data.

This study was made in a corridor with several AP’s at various distances from the source, blocking some of the beacons. A second scenario was an outdoor wide corridor with two neighbor APs, with no mayor impact on measurements. More scenarios could be added in future work to represent real and uncontrolled scenarios with no line of sight, different types of obstruction, wall types and the possibility of mobile reference points.

References

1. Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000, Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), vol. 2, pp. 775–784 (2000)
2. Yoshida, T., Wang, J.: A Position Correction Method for RSSI Based Indoor-Localization. In: Kikuchi, S., Madaan, A., Sachdeva, S., Bhalla, S. (eds.) DNIS 2011. LNCS, vol. 7108, pp. 286–295. Springer, Heidelberg (2011)
3. Xu, J.: Distance Measurement Model Based on RSSI in WSN. Wireless Sensor Network 02, 606–611 (2010)
4. Sendra, S., Garcia, M., Turro, C., Lloret, J.: WLAN IEEE 802. 11a / b / g / n Indoor Coverage and Interference Performance Study. International Journal 4, 209–222 (2011)
5. Ullah, I.: A study of Indoor positioning by using Trigonometric and Weight Centroid Localization Techniques. International Journal of Computer Science &amp 2, 60–67 (2011)
6. Lloret, J., Tomas, J., Canovas, A., Bellver, I.: GeoWiFi: A Geopositioning System Based on WiFi Networks. In: The Seventh International Conference on Networking and Services, ICNS 2011, pp. 38–43 (2011)
7. Lassabe, F., Canalda, P., Chatonnay, P., Spies, F.: Indoor Wi-Fi positioning: techniques and systems. Annals of Telecommunications - Annales Des Télécommunications 64, 651–664 (2009)
8. Kaemarungsi, K., Krishnamurthy, P.: Properties of indoor received signal strength for WLAN location fingerprinting. In: The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004, pp. 14–23. IEEE (2004)
9. Nadir, Z., Ahmad, M.: Pathloss Determination Using Okumura-Hata Model And Cubic Regression For Missing Data For Oman. In: Proceedings of the International MultiConference of II, pp. 17–20 (2010)
10. Sklar, B.: Rayleigh fading channels in mobile digital communication systems. I. Characterization. IEEE Communications Magazine 35, 136–146 (1997)
11. Bose, A., Foh, C.H.: A practical path loss model for indoor WiFi positioning enhancement. In: 2007 6th International Conference on Information, Communications & Signal Processing, pp. 1–5. IEEE (2007)
12. Chen, Y., Terzis, A.: On the implications of the log-normal path loss model. In: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, SenSys 2011, p. 26. ACM Press, New York (2011)
13. El-Kafrawy, K., Youssef, M., El-Keyi, A., Naguib, A.: Propagation Modeling for Accurate Indoor WLAN RSS-Based Localization. In: 2010 IEEE 72nd Vehicular Technology Conference - Fall, pp. 1–5. IEEE (September 2010)
14. Akl, R., Tummala, D., Li, X.: Indoor propagation modeling at 2.4 GHz for IEEE 802.11 networks. In: The sixth IASTED International Multi-Conference on Wireless and Optical Communications Wireless Networks and Emerging Technologies (2006)
15. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37, 1067–1080 (2007)
16. Luo, X., O'Brien, W.J., Julien, C.L.: Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites. Advanced Engineering Informatics 25, 355–363 (2011)

Towards Automated Extraction of Expert System Rules from Sales Data for the Semiconductor Market

Jesús Emeterio Navarro-Barrientos[1,⋆], Dieter Armbruster[1], Hongmin Li[2], Morgan Dempsey[3], and Karl G. Kempf[3]

[1] School of Mathematical and Statistical Sciences
[2] W.P. Carey School of Business
Arizona State University, Tempe, AZ, USA
{jenavar1,armbruster,hongmin.li}@asu.edu
[3] Intel Corporation, Chandler, AZ, USA
{morgan.dempsey,karl.g.kempf}@intel.com

Abstract. Chip purchasing policies of the Original Equipment Manufacturers (OEMs) of laptop computers are characterized by probabilistic rules. The rules are extracted from data on products bought by the OEMs in the semiconductor market over twenty quarters. We present the data collected and a qualitative data mining approach to extract probabilistic rules from the data that best characterize the purchasing behavior of the OEMs. We validate and simulate the extracted probabilistic rules as a first step towards building an expert system for predicting purchasing behavior in the semiconductor market. Our results show a prediction score of approximately 95% over a one-year prediction window for quarterly data.

Keywords: data mining, expert systems, probabilistic rules, semiconductor markets, system learning.

1 Description of the Problem and Conceptual Model

Forecasting sales is a huge issue for all companies and many different quantitative and qualitative forecasting methods are used by practitioners to characterize product selection, purchasing policies and predictability of purchasing behaviour of several customers [7,8,12,14]. For example, some approaches have shown that server products preferences are main memory, cache memory, brand name, CPU speed and hard disk capacity [10]. Due to the rapidly changing environment in which high-tech firms are immersed, there is a great need to understand the demands of customers and to transfer customer knowledge to product innovation [5].

[⋆] Emeterio Navarro was with the School of Mathematical and Statistical Sciences, Arizona State University, and now is with Society for the Promotion of Applied Computer Science (GFaI), Volmerstr. 3, 12489 Berlin, Germany. [email: navarro@gfai.de]

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 421–432, 2013.

Another approach consists in forecasting technology adoption and first-purchase demand using for example the Bass model [2,6,12] which describes the adoption of an innovation in the market. A different approach consists in using agent-based modeling techniques [9,13], here the usual approach is to use interdependent types of agents, for example industries, intermediate companies and consumers.

In this paper we use a qualitative data mining [3] approach to find rules from sales data that characterize the purchasing behavior in the semiconductor market. The rules are based on the following question: Which company or OEM buys which component in which period of time? For this, we calculated the probability of switching to a new product immediately or after one quarter. The novelty of our paper consists in the extraction of probabilistic rules from sales data to characterize policy decisions. To our knowledge such policy modeling has not been studied thoroughly, especially not for the semiconductor market. This presents a challenging task, specially for products characterized by different hierarchical properties, for example microprocessors, where memory architecture has usually a higher hierarchy than CPU speed. Based on the rules extracted from data, we implemented and validated an expert system to predict future sales. In this manner, our research presents an application of automated rule extraction for rule-based expert system designing for predicting future sales in the semiconductor market.

In general, in the semiconductor market the life-cycle stages of a microprocessor are [11]:

1. pre-launch stage: design of new products, production samples shipment, customer preliminary evaluation, chipset shipment for customer circuit board testing and preliminary prediction of consumer demand/supply.
2. ramp-up phase: from launching the product until 40% of estimated life-cycle or 40% of estimated market size is reached.
3. ramp-down phase: follows the ramp-up phase and ends when approx. 90% of estimated life-cycle or 90% of estimated market size is reached.
4. end-of-life: determines the end of the product life-cycle.

During the life cycle of the product different decision making strategies are present at different times for producers, providers and buyers, respectively. For example, based on the philosophy of the company and relevant external information like production costs, market position, political and economical news, different strategies are evaluated by the OEMs to determine which products to sell or buy and how many. Some OEMs may buy a new product immediately after launch into the market; some may wait and first observe other OEMs actions in the market. Some OEMs may follow a particular company philosophy in their decision making and other OEMs may arrange collaborations with strategic partners.

Fig. 1 shows the conceptual structure problem. A chip is characterized by its memory architecture (M), its platform (P), its CPU-family (C) and its CPU-speed (S). Here CPU-family is a design that works at one process technology size. Platform refers to the whole collection of components that works together on the chip. It is also referred as the chipset family. Note that changes in these characterizations come in different time intervals. Changes in memory architecture are external to Intel and happen on approximately a two-year cadence. New memories can be introduced at any time to any platform or CPU-family. The characterization of the purchasing behavior of the OEMs involves multiple overlapping products over time. In order to predict the reaction of an OEM to the introduction of a new product with a set of characterizing features (M, P, CPU, S) we need to find the rules and the hierarchies that an OEM has for buying a chip with new memories, new platforms, new CPU-families or new CPU-speeds. For example, an OEM would prefer a product with a new CPU-family and an old memory over a product with a new memory with an old CPU-family.

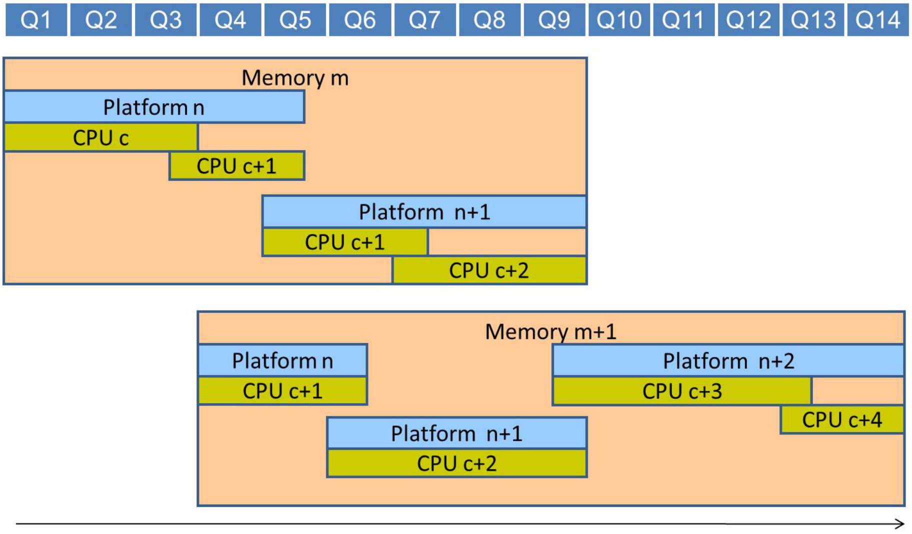


Fig. 1. Conceptual structure problem: multiple overlapping products over time

2 Data and Visual Representation

The data we used for our investigations corresponds to 269 SKUs (Stock-Keeping Unit) offered by Intel to the OEMs over 20 quarters (the exact data are not disclosed due to privacy concerns). The memory vendors choose what speeds are available, and Intel (typically) matches the Front Side Bus (FSB) to those speeds for better performance. A SKU in the laptop market leaves the factory set for one CPU-speed and one FSB/Memory speed for one platform (chipset family). For each SKU we have the following data: Memory (M) values in the interval [1,2,..8]; Platform (P) values in the interval [1,2,..7]; CPU-family (C) values in the interval [1,2,..11] and CPU-speeds (S) in GHz. For simplicity, these values are placed next to the component types, for example *M1* describes the first version of the component *memory*, *P1* describes the first version of the component *platform*, etc.

The names of the OEMs and data are encrypted/hidden. We will be referring to the OEMS as company: *A, C, D, E, F, G, H, I, J* and *O*. OEM *O (Other)* aggregates the sales of all other OEMs into a single group. A variety of OEM sizes were used, from large to small and we excluded OEM *B* from our results due to privacy concerns.

For the sake of motivation, consider the plot shown in Fig. 2 which displays the stacked volume for OEM *A* per quarter, showing choices for CPU-family *C* and Platform *P* over twenty quarters. The life-cycle stages of a microprocessor described previously can be elucidated from this figure where OEM *A* chose different volumes over time.

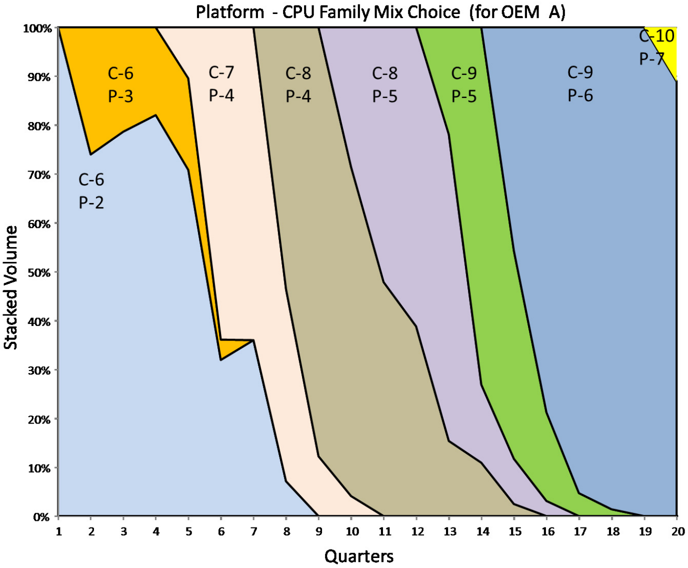


Fig. 2. Stacked volume Platform-CPU Family mix choice for OEM *A*

Moreover, the diversity of product components offered by Intel and its hierarchical structure is shown in Fig. 3 for *269* different SKUs offered by Intel during twenty quarters. Each SKU is represented by a hierarchical component tree composed by different combinations of memories *M(1,2,..,8)*, platforms *P(1,2,..,7)* and CPU-families *C(1,2,..,11)*. The CPU-speeds are located at the bottom of the tree, however, for the sake of visualization we exclude their labeling.

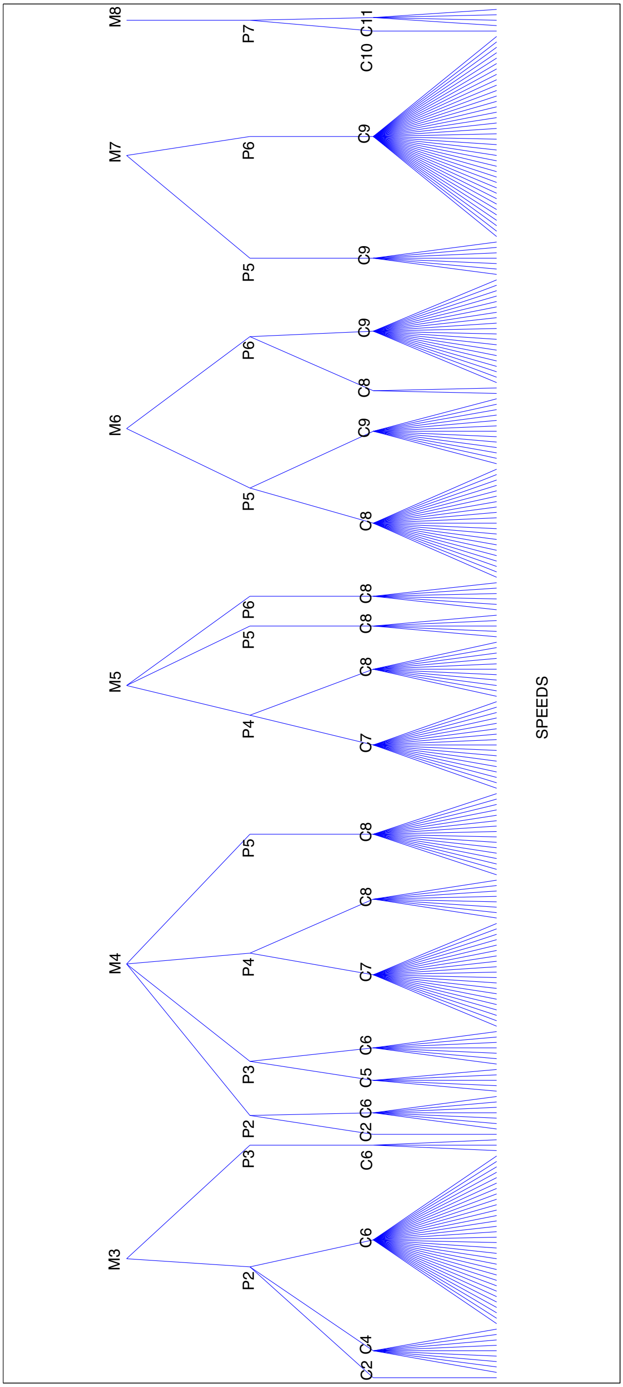


Fig. 3. All combinations of SKU-components offered by Intel in all markets

3 Rule Extraction for the Semiconductor Market

3.1 Probabilistic Rules

One of the most interesting issues for Intel is to determine how long it takes an OEM to adopt a new component/feature for a product, if at all. We extract different probabilistic rules for the adoption time of the type: *If Intel introduces a new product with new feature/component X at time $t-y$, the OEM buys that product at time t with probability $NX(t-y)$*. Note that we assume that only new products relative to the launching time by Intel are those triggering the rules.

The extraction of probabilistic rules is based on the basic measure of repeat-buying in Ehrenberg's theory [4] (the mean purchase frequency) which describes, in general, the average number of times that at least one new product component was bought during a given period of time. These probabilistic rules can be also seen as a type of fuzzy logic rule model on real data of the semiconductor market. Note that any rule is associated with a probabilistic distribution defined over the number of quarters passed since the introduction of a new product to the market.

Table 1 shows the probabilities extracted from the data for those cases where the OEMs decided to add a new feature offered by Intel to their shopping basket, independently of the fact whether other features were also launched at the same time with the same new product. Note that almost all companies add SKUs with a new feature either immediately or after a quarter. For example, OEMs D, H and O have a large probability to add a new memory (M) or a new platform (P) immediately, whereas OEMs A, C and E add a new memory or a new platform with a lag of one quarter. Now, for CPU-families (C), we can see that companies D, H, E, G, J, I and O are very reactive to new CPU-families, buying them immediately. Finally, it can be seen that OEMs D, H and O buy new features, in general, almost immediately after they are launched by Intel.

Table 1. Probabilities of adding a new feature offered by Intel. The probabilities were extracted from the OEMs purchasing behavior.

	NM(t-0)	NP(t-0)	NC(t-0)	NM(t-1)	NP(t-1)	NC(t-1)
A	0.20	0.17	0.40	0.80	0.83	0.60
C	0.20	0.17	0.50	0.80	0.83	0.50
D	1.00	1.00	0.83	0.00	0.00	0.17
E	0.20	0.33	0.75	0.80	0.67	0.25
F	0.60	0.67	0.67	0.40	0.33	0.33
G	0.60	0.67	0.71	0.40	0.33	0.29
H	0.80	0.83	0.86	0.20	0.17	0.14
I	0.60	0.66	0.71	0.40	0.33	0.29
J	0.33	0.33	1.00	0.66	0.66	0.00
O	0.80	0.83	0.20	0.20	0.17	0.17

3.2 Scoring and Rule Validation

In order to validate and test the rules extracted from the data, we built an expert system that provides the most probable microprocessor to be chosen by any OEM at any quarter. At every quarter Intel offers a set of products and the OEM uses a probabilistic decision tree to select from the current products those that better fit its needs and add those product with a new memory architecture, a new platform or/and a new CPU-family, respectively, to its shopping basket, see Fig. 4.

The resulting predictions are evaluated on the data set itself by computing a score, Eq. 1, for the match between the real product selection (from the data) by the OEMs and the products selected by a simulated OEM using the probabilistic rules in Table 1. The following scoring function was used for this purpose:

$$score(t) = \sum_{i=1}^{\mathrm{w}} \mathrm{weight}_i \frac{\mathrm{Real}(t+i) \cap \mathrm{Sim}(t+i)}{\mathrm{Real}(t+i) \cup \mathrm{Sim}(t+i)}, \tag{1}$$

where w is the window size, weight_i weights every quarter in the window according to a decaying function($\mathrm{weight_i} = \frac{\mathrm{w}-i+1}{\mathrm{w}+1}$), $\mathrm{Real}(t+i)$ is a set of product components that the real OEM selected at time $t+i$ and $\mathrm{Sim}(t+i)$ is a set of product components that the simulation of the OEM predicted to be selected at time $t+i$. The score function in Eq. 1 measures the components (without repetitions) that were correctly predicted by the simulation as a fraction of the number of all components chosen by the real OEM and the simulated OEM.

The validation of the rules for purchasing behavior in the expert system is done as follows. First, at every quarter t the basket of the OEMs is initialized with the products that the real OEM bought in that quarter. On a second step, the extracted probabilistic rules are used to determine which product would the OEM buy over the four-quarter window. Finally, the products chosen by the simulated OEM are then compared with the real selection of the OEMs using Eq. 1. The average simulation scores btained for all companies are shown in Fig. 5. Note that companies A, D, G and O are the companies with the lowest overall average simulation score. Recall that OEM O *(Other)* represents the sum of all other purchasers, therefore a low predictability for OEM O should be expected. We use the standard deviation for the error bars in the plots, however, note that the error bars with upper bound greater than one are not meaningful.

Fig. 6 shows the total average score over all OEMs for all quarters. Note that the average score is high (close to a perfect score of 100%) for most of the quarters, however, two big drop downs occurred in the middle and towards the end of the timeline. The average score over all quarters was $\langle score \rangle = 0.9392$ with an average standard deviation of $SD = 0.03085$.

Fig. 7 shows the ranked average score for purchasing-strategies using the probabilities extracted from data. Note that the purchasing-behavior of companies C, E, H, F and I can be predicted with approximately 95% precision, also note that the OEM D has the largest variance, meaning that a large uncertainty can be expected when trying to predict its behavior.

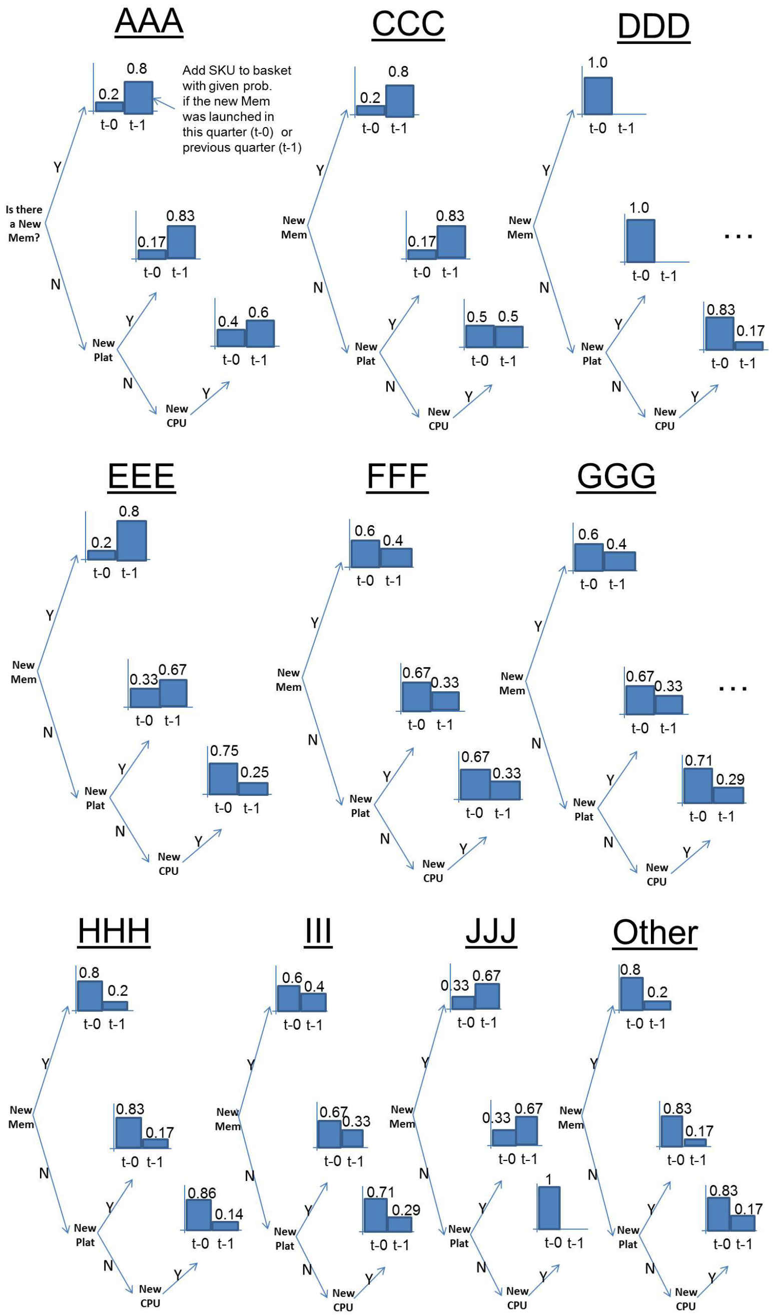


Fig. 4. Probabilistic decision trees for the purchasing behavior of OEMs in the semiconductor market. Probabilities extracted from sales data.

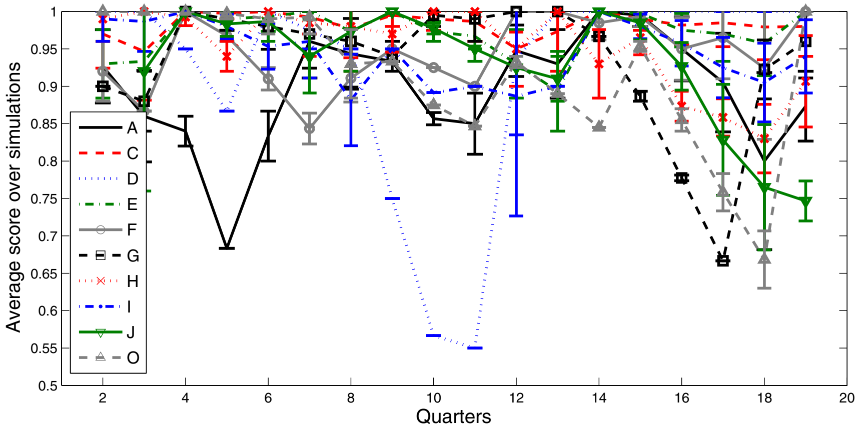


Fig. 5. Score of the prediction of OEMs purchasing behavior using probability rules on Table 1 and assuming a perfect over-horizon score

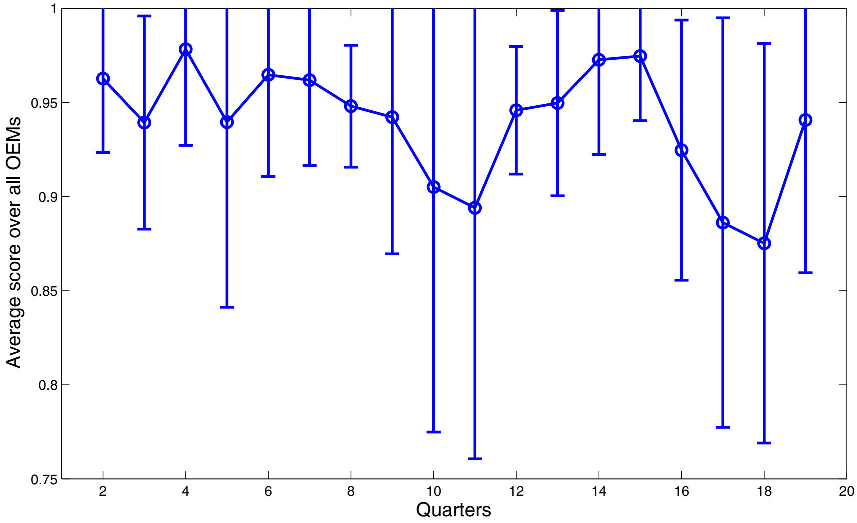


Fig. 6. Average score for all OEMs in Fig. 5 assuming a perfect over-horizon score

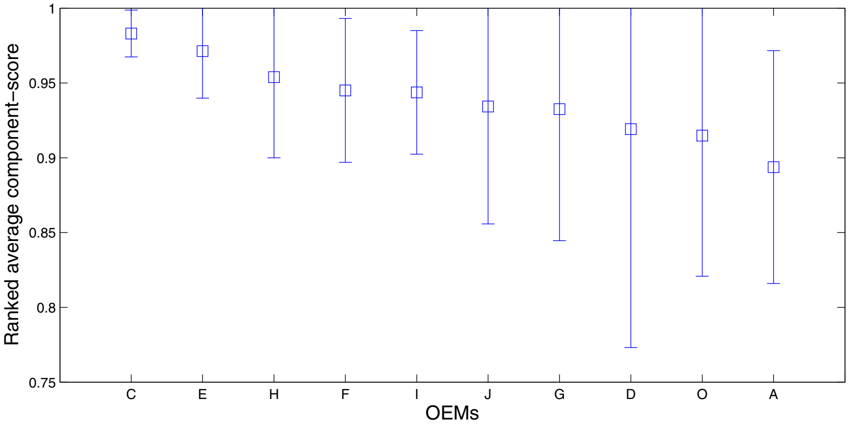


Fig. 7. Ranked average score for purchasing strategies using probabilities from Table 1

3.3 System Learning

An interesting task consists in assessing how well a system learns a given model. For this, we compare the average simulation score between the probability-based approach proposed in this paper and a *do-nothing* strategy, i.e. to keep buying at every quarter the products bought initially (the quarter before starting the four-quarter prediction window).

We used the measure *Root Mean Square Error (RMSE)* to obtain the *average system learning* of our model. For clarity, by average system learning, we mean the average distance between the strategies *do-nothing* and *do-the-best-you-can* (our probabilistic approach). For this, the *RMSE* is used as a comparison criterion between observations x_i and simulation results y_i as it is shown in the following Equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}. \quad (2)$$

Fig. 8 shows the average system learning for the simulated strategies. Note for example that even though we can predict OEM C with an average score above 95%, see Fig. 7, when we calculate the system learning for this OEM, we find that we do not learn much information from their behavior, where OEM C occupies the middle rank among the companies, see Fig. 8. Basically, this occurs because companies like OEM C, they do not tend to make big changes in its shopping basket over time. On the other hand, companies from which we do learn like OEMs G, D and H, they tend to have a more active systematic strategy.

Following these thoughts, an interesting situation consists in comparing the average system learning using RMSE with the score for the strategies, both for different window size. Note that do-nothing strategies lead to high scores because

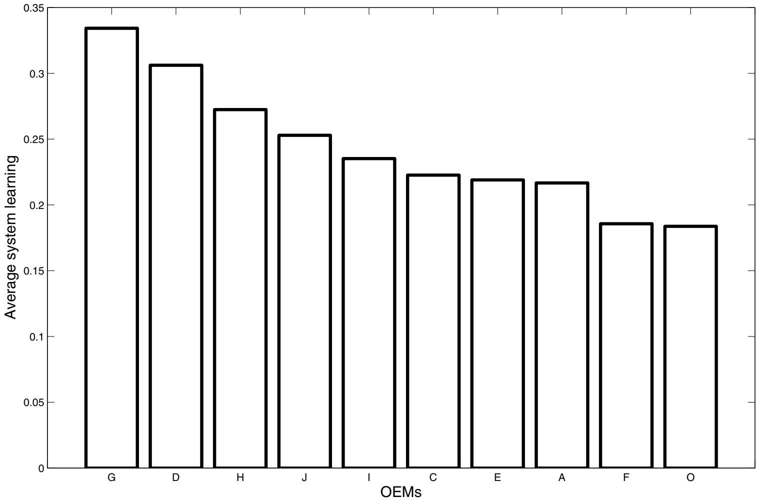


Fig. 8. Average system learning for the OEMS when using strategies based on probabilities for adding new components (Table 1), i.e. distance between *do-nothing* and *do-the-best-you-can*

every evaluation window is initialized with basket data from the real OEM. In this manner, the *RMSE* will depend on the window size because the shorter the window size, the higher the score; on the other hand, the larger the window size, the more we can learn from the OEM and the more we can predict for larger periods of time. The score will decrease if the approach used to extract the strategies from data does not model accurately the real strategy of the OEM. In our case, the window size for Figure 8 was of four quarters.

4 Conclusions and Further Work

In this paper, we analyzed sales data from the semiconductor market and extracted probabilistic rules to predict which products are chosen by the OEMs. The hierarchical characterization of product components and features in memory, platform, CPU and speed leads to different preferences among the customers. We extracted *probabilistic rules* by finding the frequency in which OEMs added features from a market basket. The probabilistic rules lead to a score of approximately 95% matching for all OEMs over a one-year prediction window.

Further work consist in investigating rules for adding/removing SKUs triggered by new cache, core or/and external factors. It would be also interesting to investigate the influence of the prediction window size in the balance between higher score and high system learning. Decision making rules based on performance/price ratios could also be investigated and compared with this probability-based approaches. Another interesting extension would be to analyze imitation/collective behavior between OEMs. A method of special interest for comparison purposes for the rule extraction could be the use of association rules [1], which are broadly used for basket marketing analysis to find products that are typically bought together. Another methods for characterizing OEMs are based on leading indicator indexes for finding business cycle patterns.

Acknowledgements. J.-E. N.-B., D.A. and H.L. gratefully acknowledge support from the INTEL Research Council. D.A. was also supported by a grant from the Volkswagen Foundation under the program on Complex Networks.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD Conference, pp. 207–216 (1993)
2. Bass, F.M.: A new product growth for model consumer durables. Management Science 15(5), 215–227 (1969)
3. Bratko, I., Suc, D.: Qualitative data mining and its applications. Journal of Computing and Information Technology 3, 145–150 (2003)
4. Ehrenberg, A.S.C.: Repeat-Buying: Facts, Theory and Applications, Griffin, London (1988)
5. Lin, R.-J., Che, R.-H., Ting, C.-Y.: Turning knowledge management into innovation in the high-tech industry. Industrial Management & Data Systems 112(1), 42–63 (2012)
6. Mahajan, V., Muller, E., Bass, F.M.: New product diffusion models in marketing: A review and directions for research. The Journal of Marketing 54(1), 1–26 (1990)
7. Mishra, P., Padhy, N., Panigrahi, R.: The survey of data mining applications and feature scope. Asian Journal of Computer Science and Information Technology 2(4), 68–77 (2012)
8. Sanders, N.R., Mandrot, K.B.: Forecasting Practices in US Corporations: Survey Results. Interfaces 24(2), 92–100 (1994)
9. Schwoon, M.: Simulating the adoption of fuel cell vehicles. Journal of Evolutionary Economics 16(4), 435–472 (2006)
10. Shao, B., Fernandez, E.: Hedonic price analysis of the server appliance market. In: Proceedings of the Seventh Americas Conference on Information Systems, AMCIS, pp. 1791–1796 (2001)
11. Wu, S.D., Aytac, B., Berger, C.A., Armbruster, C.A.: Managing short-lifecycle technology products for agere systems. Interfaces 36(3), 234–247 (2006)
12. Wu, S.D., Kempf, K.G., Atan, M.O., Aytac, B., Shirodkar, S.A., Mishra, A.: Improving new-product forecasting at Intel Corporation. Interfaces 40(5), 385–396 (2010)
13. Zhang, J.: Growing silicon valley on a landscape: An agent-based approach to high-tech industrial clusters. Journal of Evolutionary Economics 13, 529–548 (2003)
14. Zotteri, G., Kalchschmidt, M.: Forecasting practices: Empirical evidence and a framework for research. International Journal of Production Economics 108(1-2), 84–99 (2007)

Bayesian Networks for Micromanagement Decision Imitation in the RTS Game Starcraft

Ricardo Parra and Leonardo Garrido

Tecnológico de Monterrey, Campus Monterrey
Ave. Eugenio Garza Sada 2501. Monterrey, México
{a00619071,leonardo.garrido}@itesm.com

Abstract. Real time strategy (RTS) games provide various research areas for Artificial Intelligence. One of these areas involves the management of either individual or small group of units, called micromanagement. This research provides an approach that implements an imitation of the player's decisions as a mean for micromanagement combat in the RTS game Starcraft. A bayesian network is generated to fit the decisions taken by a player and then trained with information gather from the player's combat micromanagement. Then, this network is implemented on the game in order to enhance the performance of the game's built-in Artificial Intelligence module. Moreover, as the increase in performance is directly related to the player's game, it enriches the player's gaming experience. The results obtained proved that imitation through the implementation of bayesian networks can be achieved. Consequently, this provided an increase in the performance compared to the one presented by the game's built-in AI module.

Keywords: Bayesian Networks, RTS Video Games, Intelligent Autonomous Agents.

1 Introduction

The video game industry has been in constant development along the latest years. Numerous advancements have been made on graphics improvement in order to produce more realistic game environments, as well as the hardware that is required to handle these upgrades. The research area known as Artificial Intelligence (AI) has also its important place in the video game industry, among developers and researchers. AI is applied, on most part, to those units or characters that are known as non-playable characters (NPCs) in order to generate behaviours and as a medium for making decision, either autonomously or collectively. NPCs, as its name express, are all of the units on every game that are not controlled or modified by any player. A variety of algorithms, learning methods and reactive methods have been applied in games to provide a better gaming experience to the players.

Real time strategy (RTS) games are a stochastic, dynamic, and partially observable genre of video games [2]. Due to their strategic nature, RTS games require to continuously update the decisions, either from a player or the game's AI

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 433–443, 2013.

module. These type of games require throughout planning of strategies according to available information they have access to, such as disposable resources, available units and possible actions. The RTS game that was used along this research is titled "Starcraft: Broodwar" [3].

In the RTS game Starcraft, each player can posses up to a population equivalent to 200 units. Hence, the management of units can be divided into two segments: macromanagement and micromanagement. Macromanagement is usually regarded as the highest level of planning. It involves the management of resources and units productions. Micromanagement, on the other hand, refers to the individual control of units on combat. Micromanagement applied to a player's game require a large amount of time and precision.

Performing the necessary micromanagement tactics on 200, 100 or even 50 units can become a challenge that many players can't overcome. Hence, a method or process is required in order to continuously generate micromanagement decisions along the game or combat. Nevertheless, given that the environment provided by a RTS game can be partially observable, an approach that can cope with uncertainty is required. Moreover, if the output of the process can be match to an individual's personal strategy, the performance on a full scale game could be higher.

Considering the environment is partially observable, a bayesian network approach was implemented. Bayesian networks (BN) are one of the most appealed techniques to learn and solve problems while coping with uncertainty. BNs use probability and likelihood that specific success of events given a set of evidence. Moreover, it represents its model through compact acyclic directed graphs (DAG).

This paper exploits two vital tools that provides both graphical model and coded libraries, GeNIe and SMILE respectively, developed by the Decision Systems Laboratory of the University of Pittsburgh [4]. The GeNIe (Graphical Network Interface) software is the graphical model for the portable Bayesian network inference engine SMILE (Structural Modelling, Inference, and Learning Engine). GeNIe provides a graphical editor where the user can create a network, modify nodes properties, establish the conditional dependencies and much more. SMILE are a platform independent C++ libraries with classes supporting object oriented programming.

2 Related Work

In the latest years, video games have been exploited as research platforms for AI. The implementation of bayesian networks on games, as well as different approaches to micromanagement have capture the attention of different researchers. In Cummings et al.[5], a bayesian network approach for decision making was implemented in order to determine which chest to open out of two possible options. They use information recorded about previous open chests in order to calculate the probabilities for the CPT's of the nodes.

There has been previous work where bayesian networks have been implemented in RTS games. In their work, Synnaeve and Bessiere[6]worked under the assumption that decisions made by human players might have a mental probability model about opponents opening strategies. They used a data set presented on Weber and Mateas[7] work, containing 9316 Starcraft game logs and applied it to a bayesian model they proposed for opening recognition. They implement a backtracked model that helped them cope with the noise produced from missing data. The results involved a positive performance, even in the presence of noise due to removal of observations.

Further work of Synnaeve and Bessiere [8] involves unit control for the RTS game Starcraft. They propose a sensory motor bayesian framework for deciding the course of actions and the direction it should take. They make use of variables such as possible directions, objective direction and include other variables that represent damage, unit's size and priorities. They also applied potential fields that influence the units depending on their type and goal for complementing the decision making.

Moreover, more research related to micromanagement has been made in the work presented by Szczepanski and Aamodt[9]. They applied case based reasoning to the RTS game Warcraft3 for unit control. The case based reasoning received the game's state in order to retrieve the decisions of the most similar case and adapt it. Then, the unit receives a command with the information on how to respond to that specific case. This process is repeated every second during the experiment. The results from their experiments describe a consistent increase on the performance displayed by the AI against its opponents.

3 Bayesian Networks

The Bayesian Networks (BN), also referred as belief networks, are directed graph models used to represent a specific domain along with its variables and their mutual relationships. They make use of the Bayes' Theorem in order to calculate the probability of a specific event given known information, usually referred to as evidence. Bayes' Theorem is expressed as:

$$P(E|H) = \frac{P(H|E)P(H)}{P(E)} \qquad (1)$$

BN are widely used in order to cope with uncertainty at a reasoning process at discrete sample space. BN use inference based on the knowledge of evidence and the conditional relationship between its nodes. Their implementation is included in decision theory for risk analysis and prediction in decision making. These networks make use of the bayesian probability, which was described by Heckerman [10] as a person's degree of belief on a specific event.

The bayesian networks contain nodes $(X_i..X_j..X_n)$ in order to represent the set of variables that are considered to influence the domain. The values they hold might either be discrete or continuous. Nodes are wired together by a series of direct connections represented by links or arcs. These arcs represent the relationship of the dependencies between variables. Hence, this networks are called directed acyclic graphs (DAGS). Further information regarding bayesian networks is presented by Korb [1] and Russell [2].

4 Bayesian Networks for Decision Imitation

A modeling process was established in order to imitate the decisions taken by the player. This process was applied to the experimental scenario described later on the paper. The process can be broken down into three different segments: Information extraction from the player, bayesian network model creation, and bayesian network prediction. Figure 1 illustrates the steps followed during experimentation.

The first step involves information extraction from player. A human player is briefed about the experiment. The human then plays on specific map for n number of cycles, where n is equal to 30. Meanwhile, the environment is sensed and a data log with the state of relevant variables of the environment is generated. There are a total amount of n logs generated. After the cycles are done, the logs are merged together to form a database, which will be used for the training of the bayesian network.

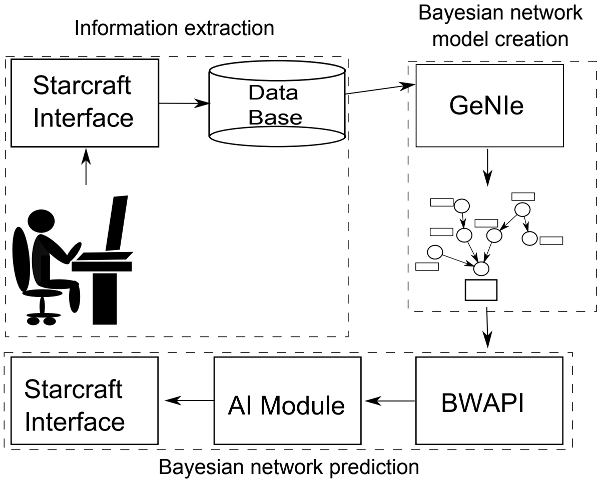


Fig. 1. Implementation Overview

The next step is the bayesian network model creation. The model is generated with the software known as GeNIe. Initially, we generate the nodes that represent the variables from the database that will be required for the decision making process. In order to select those nodes we queried an expert, the player whose decisions are being imitated, about the variables that influence his decision.

Then, the relationship between variables is sought and defined through arcs on the model. After the model is defined, the conditional probability tables are then filled with the calculated probabilities. These probabilities are obtained by loading the data extracted from the player and performing GeNIe's built-in parameter learning.

The last step involves the implementation of the bayesian network prediction. The experimental map is loaded once more with an modified Starcraft AI module that make use of the bayesian network. The environment is sensed by the AI module and the obtained information from the variables are used as evidence on the bayesian network. Once the propagation is made, we choose the action, direction, and distance with the highest posteriori probability as the one to be implemented. The sensing process and execution process are both made every five game frames. Given the game is running at the fastest speed, these processes are done, approximately, every one fifth of a second.

5 Experimental Setup

The Broodwar Application Programming Interface (BWAPI)[11] was used in order to implement the bayesian networks in the game. It was through it, as well, that a link between the game and the bayesian network's library was created. BWAPI generates a dynamic library (DLL) that can be loaded into Starcraft: Broodwar and enable a modified Starcraft AI module to be used.

The map used for the experiments was created on the Starcraft: Campaign Editor. The fighting area is composed of a 13 x 13 tiles diamond shaped arena. The fog of war, which forces the environment to be partially observable, was enabled. This limit the information available by the player's unit to that available on their field of vision. This property is also kept when the Starcraft AI module that contains bayesian network is used.

The units controlled by the player are situated in the middle of the arena while the opponent enemy forces are situated on the bottom part of it. Once the map is loaded, there is a 5 second time window for the player to select his units and establish them as hotkeys for better performance. After the time is spent, the enemy forces attack the player's units. The game ends when either forces is left without units.

5.1 Scenario: 2 vs. 3

In the scenario, we deployed two Dragoons from Protoss race and three Hydralisks from the Zerg race. The two Dragoons deployed by the player can defeat the three opponent's Hydraliks without any other need than a micromanagement well done. If it is not performed correctly, the game will end in the player been defeated.

The data was extracted from the interaction of the player with the setup previously mentioned on the foretold map. The information is extracted from the game every five frames throughout the player's game. The variables obtained for this scenario are declared on Table 1. The resulting database used for the training contains data from thirty repetitions. This scenario's training database contains over 8000 instances.

Table 1. Variables for 2 vs 3 Scenario

Friend1_HP	Next_Action
MyHP	
Attacked	
CurrentTargetDistance	
Next_Action	NextTargetDirection
Friend1_Direction	
Enemy1_Direction	
Enemy2_Direction	
Enemy2_Distance	NextTargetDistance
Enemy1_Distance	

We tried to keep the bayesian network as simple as it could without compromising the performance. Hence, we consulted the expert in order to design the corresponding network. The resulting network generated with the player's aid is illustrated in Figure 2. According to the player, the unit should decide his next action depending on his hitpoints, his ally hitpoints, whether it is targeted by an enemy or not, and the distance to his current target. The direction of the action to be done by the controlled unit depends on the next action as well as the direction of his ally and his enemies. In the scenario, a unit must consider the direction of his ally so they do not collied when they move to a safer place. Finally, the distance at which the action would be made is defined by the distance toward the enemies. Regardless the presence of a third enemy, the decision can be made by considering two enemies. Hence, it can be observed on the proposed network the lack of the third enemy.

6 Results

The results obtained from the scenario are presented in this section. First, we present the performance contrast between the Starcraft built-in AI module and the AI module containing the bayesian network. Then, a comparison between the decisions taken by the player and the decisions taken by the bayesian network AI module is made. In this comparison, a set of variables are selected and set to a specific discretized value. Moreover, the possible output corresponding to those values is graphed and compared.

The table declares the probability of choosing a specific next action, first column, given the available evidence, first two rows. In can be observed on the presented tables that the action chosen by the AI module resembles the decision taken by the player. This process was repeated for a series of other configurations of available information. The results are presented on Table 6.

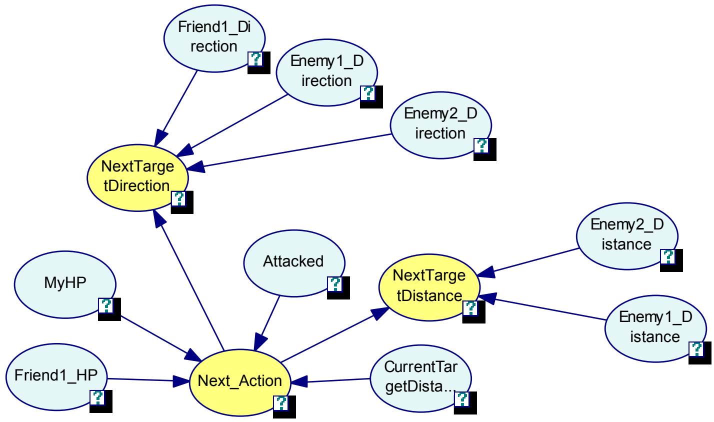


Fig. 2. Bayesian network model for a two Dragoons vs three Hydralisks

6.1 Scenario: 2 vs. 3

The first part of the results is a comparison between performances of the Starcraft built-in AI module and the module that implements the player's decisions. We elaborated 200 games for each AI module and obtained whether they achieved victory or not. There was a significant difference in performance between the AI modules. As express previously on the scenario's setup, the built-in AI module is not capable of defeating the opposing units. This caused that the built-in AI module generated 0% of victories. Nonetheless, if a bayesian network is used in order to imitate a player's micromanagement decisions, the expected percentage of victory increase to 44.5%.

The second part of analysing the results requires a comparison between the expected decisions according to the player's information and the decisions taken by the bayesian network AI module. We generated a series of tables that contains the probability distribution for a specific set of evidence on the game. We compared the course of action taken by the player, the training set data, with the one taken by the bayesian network AI module, the test set data. Both sets of data will be presented as tables representing specific environments and their prediction.

Table 2 contain the training distribution presented by the player, while Table 3 contain the distribution presented by the Starcraft AI module that implemented the bayesian network. The first column of the tables refer to the variable to be predicted and the corresponding states it contain. The states of the node are declared on rows on the first column. The rest of the columns in the tables express the combination the state or set of states established as evidence.

In Table 2 and Table 3 the variable to be predicted, labelled in the first column, is NextAction. The states of NextAction considered for the comparison are AttackMove, AttackUnit, PlayerGuard and Move. The rest of the columns establish the combination of specific data as evidence. Table 2 and Table 3 have

MyHP variable set as Medium, FriendHP set as Full, Attacked set as True and the distance to the current target,CurrentTargetDistance, with several possible values: Melee,Ranged1, Ranged2, and Ranged3. Therefore, all of tables presented declare the variable intended for prediction as well as the variables and states that are being established as evidence.

Table 2. Player's decisions over NextAction considering MyHP = Medium, Friend1_HP = Full, Attacked = True and CurrentTargetDistance

	Medium			
	Full			
	True			
Next Action	Melee	Ranged1	Ranged2	Ranged3
AttackUnit	66.67%	5.26%	0.00%	90.00%
Move	11.11%	84.21%	100%	5.00%
PlayerGuard	22.22%	5.26%	0.00%	5.00%
AttackMove	0.00%	5.26%	0.00%	0.00%

Table 3. Bayesian network AI module's decisions over NextAction considering MyHP = Medium, Friend1_HP = Full, Attacked = True and CurrentTargetDistance

	Medium			
	Full			
	True			
Next Action	Melee	Ranged1	Ranged2	Ranged3
AttackUnit	100%	0.00%	0.00%	97.80%
Move	0.00%	100%	100%	0.49%
PlayerGuard	0.00%	0.00%	0.00%	1.71%
AttackMove	0.00%	0.00%	0.00%	0.00%

Further comparison was made with the NextTargetDirection node. The results obtained on the experiments are encouraging. Table 4 presents the probability distribution of the training data in two different situations. Table 5 contains the probability distribution of the obtained performance of the modified Starcraft AI —module. Hence, it can be observed that the selection of the direction an action must be done resemble the selection observed on the player's data.

Finally, we present on Table 6 an overview of the behavior of correct and incorrect decision made by the bayesian network AI module. Correct decision refers to the match of the state of a variable with the highest probability between the player's decisions and the AI module decisions, such as the ones presented

Table 4. Player's decisions over NextTargetDirection considering NextAction = AttackUnit, Friend1_direction = Region5, Enemy2_direction, and Enemy1_direction

	AttackUnit	
	Region5	
	Region4	Region1
Next Target Direction	Region3	Region1
Region1	0.00%	92.68%
Region2	0.00%	3.41%
Region3	0.00%	1.46%
Region4	100%	0.00%
Region5	0.00%	0.98%
Region6	0.00%	1.46%

Table 5. Bayesian network AI module's decisions over NextTargetDirection considering NextAction = AttackUnit, Friend1_direction, Enemy2_direction = Region1, and Enemy1_direction = Region1

	AttackUnit	
	Region5	
	Region4	Region1
Next Target Direction	Region3	Region1
Region1	0.00%	96.80%
Region2	0.00%	0.00%
Region3	0.00%	2.36%
Region4	100%	0.00%
Region5	0.00%	0.00%
Region6	0.00%	0.84%

Table 6. Bayesian network AI module's decisions performance over 50 different situations

	Bayesian AI module
	Scenario 2 vs 3
Correct Decision	80%
Incorrect Decision	20%

on previous tables. Incorrect decision refer to the existence of discrepancy on the chosen state of an output between the player's decisions and the AI module decisions, given the same evidence is presented. An example of correct decision can be shown by considering Table 4 and Table 5. The correct decision refers to the match between tables where Region4 is selected by both, the player and the bayesian network AI module, given the first set of evidence.

6.2 Discussion

The proposed method was designed to imitate player's decisions in the RTS game Starcraft. By implementing belief networks we can make use of an expert's opinion, in this case a Starcraft player, in order to establish a bayesian network model that suits his decisions. Moreover, it is complemented by applying the knowledge of the player game information to obtain the conditional probabilities of the network. It can be observed on Table 6 that the correct imitation of decisions of the player's decisions done by the bayesian network AI module is done with a high accuracy rate.

The performance observed by the bayesian network AI module excels the performance obtained from the Starcraft built-in AI module. The 44.5% of victories provided by the experiments establishes the increase on it. This percentage is partially low given the attacks made by the default Starcraft AI module does not follow the same pattern every time. For example, the enemies may all attack the same unit controlled by the player, or they can split to attack both the player's units.

Hence, further scenarios were tested as well and their average performance exceeds the 60% of victories. It is clear that by introducing an external influence to the built-in Starcraft AI module an increase on performance can be made. Further research on imitating decisions can be made using the RTS game Starcraft as test-bed.

7 Conclusion

We presented a bayesian network approach for unit micromanagement in a RTS game. The results obtained in this research support the hypothesis of a performance improvement on the Starcraft built-in AI module. The importance of the increase 44.5% in victories is significant given the fact that the default performance is of 0% victories. Moreover, this method enables a performance that resembles that of the player. In a full RTS game, the advantage of having unit you controlled synchronized with you own strategies can enrich the gaming experience for the players. The results also support the fact that research on bayesian network might lead to interesting work on imitating decisions taken by humans. The bayesian networks provide a stable, understandable and transparent method to generate the decision imitation.

There is future work to be done in our research. The proposed learning method in our research is based on an offline learning. Further work can involve a dynamic updating on the belief network while the player is interacting with the game. This can enable online learning in order to train a bayesian network on a full game rather than on a specific map.

References

1. Korb, K., Nicholson, A.: Bayesian Artificial Intelligence. Chapman and Hall Editorial (2010)
2. Russell, S., Norvig, P.: Artificial Intelligence a Modern Approach, 3rd edn. Pearson Education (2009)
3. Blizzard Entertainment: Starcraft, `http://us.blizzard.com/en-us/games/sc/` (accessed in January 2012)
4. Decision System Laboratory, University of Pittsburgh: GeNIe & SMILE, `http://genie.sis.pitt.edu/` (accessed January 2012)
5. Cummings, J.: Bayesian networks in video games. Pennsylvania Associataion of Computer and Information Science Educators (2008)
6. Synnaeve, G., Bessiere, P.: A Bayesian Model for Opening Prediction in RTS Games with Application to StarCraft. In: IEEE Conference on Computational Intelligence and Games (2011)
7. Weber, B.G., Mateas, M.: A Data Mining Approach to Strategy Prediction. In: 2009 IEEE Symposium on Computational Intelligence and Games (2009)
8. Synnaeve, G., Bessiere, P.: A Bayesian Model for RTS units control applied to StarCraft. In: IEEE Conference on Computational Intelligence and Games (2011)
9. Szczepanski, T., Aamodt, A.: Case-based reasoning for improved micromanagement in Real-time strategy games (2008)
10. Heckerman, D.: A tutorial on learning with Bayesian Networks. Microsoft Research, Advanced Technology Devision: Microsoft Corporation, US (1995)
11. BWAPI - An API for intereacting with Starcraft: Broodwars (1.16.1), `http://code.google.com/p/bwapi/` (accessed in January 2012)

Designing and Implementing Affective and Intelligent Tutoring Systems in a Learning Social Network

Ramón Zatarain-Cabada[1], María Lucía Barrón-Estrada[1], Yasmín Hernández Pérez[2], and Carlos Alberto Reyes-García[3]

[1] Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán Sinaloa, 80220, México
[2] Instituto de Investigaciones Eléctricas, Cuernavaca, Morelos, México
[3] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) Luis Enrique Erro No. 1, Sta. Ma. Tonanzintla, Puebla, 72840, México
{rzatarain,lbarron}@itculiacan.edu.mx, myhp@iie.org.mx, kargaxxi@inaoep.mx

Abstract. In this paper we present step by step the design and implementation of affective tutoring systems inside a learning social network using soft computing technologies. We have designed a new architecture for an entire system that includes a new social network with an educational approach, and a set of intelligent tutoring systems for mathematics learning which analyze and evaluate cognitive and affective aspects of the learners. Moreover, our intelligent tutoring systems were developed based on different theories, concepts and technologies such as Knowledge Space Theory for the domain module, an overlay model for the student module, ACT-R Theory of Cognition and fuzzy logic for the tutoring module, Kohonen neural networks for emotion recognition and decision theory to help students achieve positive affective states. We present preliminary results with one group of students using the software system.

Keywords: affective computing, intelligent tutoring systems, neural networks, fuzzy systems.

1 Introduction

Emotions are human feelings associated with mood, temperament, personality, disposition, and motivation and also with hormones such as dopamine, noradrenaline, and serotonin. Motivations direct and energize behavior, while emotions provide the affective component to motivation, positive or negative.

Emotions are prominent elements always present in the mind of human beings [1]. Expressing emotions during oral communication is a way to complement information about the speaker. This information complements the information contained in the explicit exchange of linguistic messages. Paul Ekman leaded studies on the effect of emotions in the speech and their relation to facial expressions. More recently computer scientists have got involved in the problem of automatic emotion

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 444–455, 2013.

recognition and have classified emotions using pattern recognition techniques. Knowing the emotional state of individuals offers relevant feedback information about the psychological state of a speaker in order to take important decisions on how a system's user should be attended. Automatic emotions recognition can improve the performance, usability and in general, the quality of human-computer interaction systems, students learning productivity, client attention systems and other kinds of applications.

In recent years, intelligent tutoring systems (ITSs) have incorporated the ability to recognize the student's affective state, in addition to traditional cognitive state recognition. These ITSs have special devices or sensors to measure or monitor actions facial, skin conductance, speech features, etc. and as a result recognize the emotional or affective student [2, 3, 4]. However, the area of affective computing applied to the STI is still developing this being due to the fact that ITSs are intended to measure cognitive rather than affective states [5, 6, 7, 8, 9].

Research on affective computing includes detecting and responding to affect. Affect detection systems identify frustration, interest, boredom, and other emotions [2, 3]. On the other hand, affect response systems transform negative emotional states (frustration, boredom, fear, etc.) to positive ones [10, 11].

In this paper we present the architecture and the implementation of an affective and intelligent tutoring system embedded in a learning social network. Both elements, the ITS and the social network, is going to be used to improve poor Math results in the ENLACE test (National Assessment of Academic Achievement in Schools in Mexico). ENLACE is the standardized evaluation of the National Educational System, applied to students in Grades 1-9 in public and private schools. The results of ENLACE applied in early 2011 to 14 million children from third to ninth elementary level, reveals that more than nine million students have an "insufficient" and "elemental" level in learning mathematics (http://www.enlace.sep.gob.mx/). This test is in Spanish and measures learning in math, Spanish, and a third subject that changes every year.

The paper's organization is as follows: in Section 2, we describe the System Architecture. In Section 3 we present the main structure of the affective and intelligent tutoring system. Results are shown in Section 4 and conclusions and future work are discussed in Section 5.

2 Social Network and ITS Architecture

The Learning Social Network has the basic functionalities in all social networks, but its main feature is that it includes an ITS that offers the course content in a personalized style to users, as shown in Figure 1.

Users of the network are students, parents, and teachers where students are associated with personal, academic and affective information stored in a student profile. The profile contains initial information of the student (e.g. personal and academic information). The profile also stores information collected dynamically during navigation of the student in the social network and ITS. Some of this information includes the student performance on tests and his/her affective or emotional states. The student's cognitive states are obtained from academic results (tests) during the visit of the respective ITS. The students' affective states are inferred by using sensors and neural networks that are supervising users' emotions. Once the affective student state has been determined, the affective module has to respond accordingly. To do that, the tutor uses a dynamic decision network which establishes parameters that enable a mapping from the affective and cognitive student state to tutorial or affective actions. The intelligent tutorial actions then are composed by a cognitive and an affective component obtained from problems solved by the student and from the emotions.

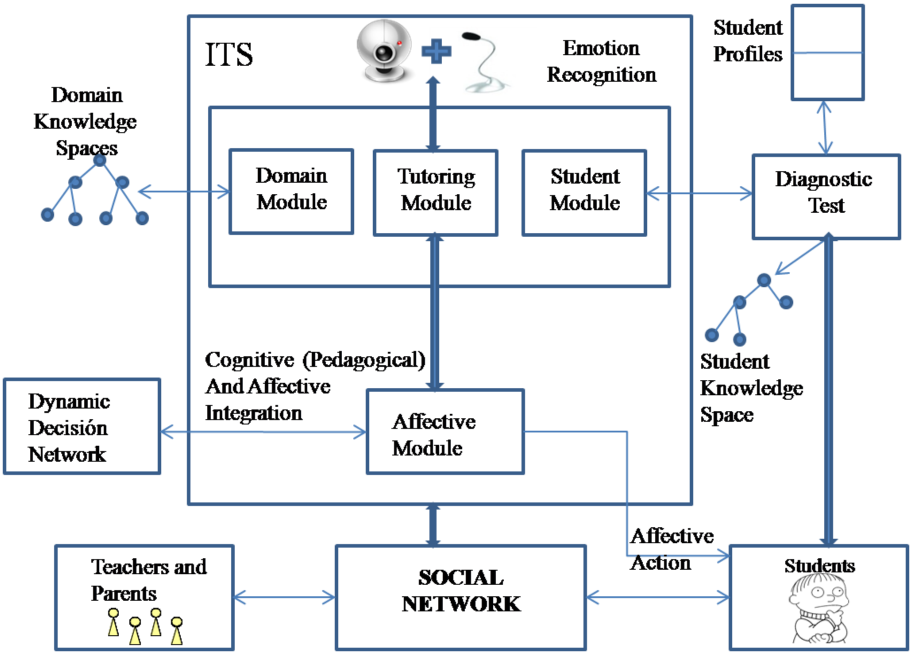


Fig. 1. Social Network and ITS Architecture

3 The Affective ITS

The affective and intelligent tutoring system adopts the traditional trinity model known as the four-component architecture where a user interface has access to three main modules: domain, student, and tutoring modules. In Figure 1 we can observe the main components of this ITS: the domain module, the tutoring module, the student module, and the affective module.

Domain Module: The knowledge acquisition and representation for the domain or expert module is a major problem which we handle through different concepts related to Knowledge Space Theory [12]. This theory provides a sound foundation for structuring and representing the knowledge domain for personalized or intelligent tutoring. It applies concepts from combinatory theory and we use it to model particular or personalized tutors according to different cognitive styles.

A course can be seen as a discipline-specific knowledge space (a particular tree diagram) containing chapters which in turn are made by subjects. The total of nodes in the tree represents the expert knowledge. The domain module is stored as a XML document whose structure is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<domain>
  <name> Domain Name  </name>
  <chapters>

    <chapter  id="1">Chapter1 </chapter>
    <urlDomain>Domain/Chapter1.xml</urlDomain>
    <urlTest>Domain/Test_Chapter1.xml</urlTest>

    <chapter  id="2">Chapter2 </chapter>
    <urlDomain>Domain/Chapter2.xml</urlDomain>
    <urlTest>Domain/Test_Chapter2.xml</urlTest>
    ……
  </chapters>
</domain>
```

Each chapter has an associated URL, which localizes the chapter specific content and the diagnostic test of the student module.

Student Module: This module is responsible for assessing the student performance to establish their cognitive abilities and reasoning skills. It provides the information about student competencies and learning capabilities. The module identifies what the student's knowledge is through a diagnostic test. The test results show what the student knows and what he needs to learn. The student module can be seen as a subset (sub-tree implemented) of all knowledge possessed by the expert in the domain (module) and a student profile stores this, as shown in the right part of figure 1. The representation is based on a model called "Overlay", where the student's knowledge is a subset of the expert knowledge. As the student uses the intelligent tutoring system he/she expands this subset [13]. For every student there is a static profile, which stores particular and academic information, and a dynamic profile, which stores information obtained from the student navigation and from emotion recognition.

When a student first accesses the ITS, he/she has to answer the diagnostic test, which allows the construction of the student knowledge space. In the diagnostic test each question has a ranking, and depending on this, we give different weights to the answers. Most difficult questions worth 3 points, intermediate questions worth 2 points and easy ones worth 1 point.

For the student's grade we use the following formula:

Student's Grade = Total Points scored / Sum of points in the questions

After the test results, a small algorithm establishes the level of student learning and the teaching method. Next, we present the algorithm for assigning the learning level and the teaching method for a student learning the topic of multiplications. The method is chosen according with the official Math program in Mexico’s Public School.

```
If (Grade < 4) Then
        Learning level = easy;    Method = Lattice
Else
If (Grade < 5) Then
        Learning level = easy;    Method= traditional
Else
If (Grade < 9) Then
        Learning level = Normal;    Method = traditional
Else
        Learning level = Difficult;    Method = Traditional
```

For the student's knowledge representation we use two categories: **Topics** where each time the student takes a topic, we store the history of subjects; and **student experience** that stores the history of grades by subject. Both are based on the Overlay model which allows us to know the subset of knowledge that the student knows.

Tutoring Module: The tutoring module was implemented based on ACT-R Theory of Cognition [7]. We implemented production rules (procedural memory) and facts (declarative memory) via a set of XML rules. Furthermore, we developed a new knowledge tracing algorithm based on fuzzy logic, which is used to track student's cognitive states, applying the set of rules (XML and Fuzzy rules) to the set of facts. The benefit of using fuzzy rules is that they allow inferences even when the conditions are only partially satisfied. Next, we show a part of the set of production rules for multiplication operations, being the rules written in XML format:

```
<multiplicacion>
    <problemas tipo="dificil" nProblemas="...">
      <p1>
        <multiplicando>17345</multiplicando>
        <multiplicador>9</multiplicador>
        <resultados>
          <r1>45</r1>
          <r2>40</r2>
          <r3>31</r3>
          <r4>66</r4>
          <r5>15</r5>
        </resultados>
      </p1>
      ...
    <p...>
      <multiplicando>748392</multiplicando>
```

```
        <multiplicador>7</multiplicador>
        <resultados>
            <r1>14</r1>
            <r2>64</r2>
            <r3>27</r3>
            <r4>58</r4>
            <r5>33</r5>
            <r6>52</r6>
        </resultados>
    </p…>
</problemas>
```

Each topic divides the problem into different categories from 1 to n. The tutoring system reads the rules, and presents the exercises according to a level of difficulty in the problem. The student cannot move to the next state (input) unless he solves correctly all the exercises. During this transition he can ask for help and in case of mistake, an error message is displayed to help discover what the correct answer is. Once the student completes the exercise, the student profile is updated with information on the type and difficulty level of the exercise, as well as the amount of mistakes, assistances, and the time it took to solve the exercise. These variables (Difficulty, assistances, errors and time) will be required to decide the next exercise the student will take. For this we implemented a Fuzzy Expert System that eliminates arbitrary specifications of precise numbers and creates smarter decisions, taking into account a more human reasoning. Fuzzy sets are described in table 1 and figure 2 for output variable Difficulty: shows four fuzzy rules which establish the degree of difficulty of the next student's problem.

Table 1. Fuzzy Values for Variable Difficulty

	Difficulty (%)	**Normalized Values**
Very Easy	0% - 10%	0 – 0.1
Easy	0% - 30%	0 – 0.3
Intermediate	20% - 80%	0.2 – 0.8
Difficult	70% - 100%	0.7 – 1.0
Very Difficult	90% - 100%	0.9 – 1.0

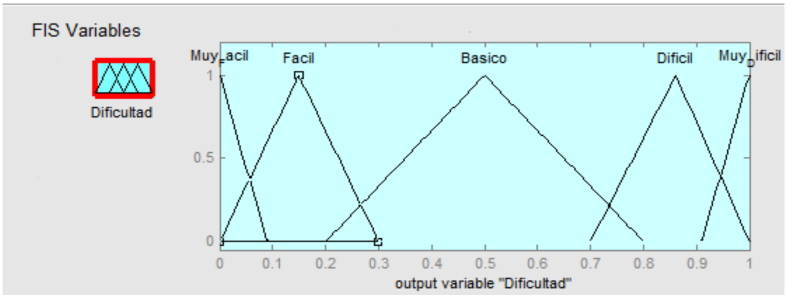


Fig. 2. Fuzzy Sets for Variable Difficulty

Table 2. Fuzzy Rules

Regla 1	If(Error is Poco) and (Ayuda is Poca) and (Tiempo is Muy_Rapido)then (Dificultad is Muy_Dificil)
Regla 2	If(Error is Poco) and (Ayuda is Poca) and (Tiempo is Rapido)then (Dificultad is Muy_Dificil)
Regla 3	If(Error is Poco) and (Ayuda is Poca) and (Tiempo is Normal)then (Dificultad is Muy_Dificil)
Regla 4	If(Error is Poco) and (Ayuda is Poca) and (Tiempo is Lento)then (Dificultad is Muy_Dificil)

Affect Recognition and Handling: Emotions are detected by expression of the face and by speech. The method used for the detection of visual emotions is based on Ekman's theory [14], which recognizes ten emotions. We only recognize 7 emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. To determine the emotion in the face we follow three steps: The first step is to detect people's faces in an image. We use the method in [15], which uses the template of P. Sinha [16,17]. The only difference in our method is instead of using a back-propagation neural network, we use a Kohonen neural network, with 20x20 neurons for the input and 2 neurons for the output (to determine if a face is detected or not). The second step is the extraction of facial features. Once we have identified a face within the image (or in the video), we get some points or features that allow us to minimize the amount of data to be used as input to the neural network, but always leaving the data or points that are necessary to identify the expression of an emotion. In the algorithm of [15], they get 20 points and the Euclidian distance between the points. We add more points, as used in [18], but also dividing the face in upper face features and lower face features. The third step is to enter data into the neural network.

Automatic emotion recognizer from speech is supported by the fact that changes in the autonomic nervous system indirectly alter speech, and use this information to produce systems capable of recognizing emotions based on extracted features of speech. For example, speech produced in a state of fear, anger or joy becomes faster, louder, precisely enunciated with a higher and wider pitch range. Emotional speech processing recognizes the user's emotional state by analyzing speech patterns. Vocal parameters and prosody features such as pitch variables and speech rate are analyzed through pattern recognition [19, 20].

Because fuzzy logic has shown good results in problems where the information is complex, for this work we chose ANFIS, which combines fuzzy logic and neural networks capabilities, to carry on the recognition of emotions from speech. We combine the advantages of ANFIS with the optimizing behavior in the search of solutions of Genetic Algorithms. We try to take advantage of the capacity of ANFIS to implement non linear mapping from input patterns towards the corresponding emotional state, and the optimizing capability of genetic algorithms to find the best features subset and the best configuration parameters to create the most adequate ANFIS architecture. The parameters to optimize are: number of membership functions, type of input membership functions and type of output function. In this work a Genetic Algorithm is applied to search the best parameters combination for the ANFIS architecture as well as to search the best feature combination.

Three groups of features are proposed, namely; Utterance Timing, Utterance Pitch Contour, and Voice Quality. Among the features extracted and to form the initial feature set we include; Speech Rate, Pitch, along with some of its statistic measures like: Pitch Average, Pitch Range, Median Pitch, Pitch Standard Deviation, Minimum Pitch Point, and Maximum Pitch Point. We include also features related to voice quality, like; Intensity Average, Jitter, Shimmer, Number of Voice Breaks, Degree of Voice Breaks, Harmonicity, and Voice Articulation. Acoustic features are extracted with Praat [21].

The feature set used to train the ANFIS model, as well as the ANFIS parameters settings, are coded in a binary chromosome in the following way: 4 bits to choose each of four features, making a total of 16 bits. Other parameters for the ANFIS configuration are also coded in the same chromosome as follows; 2 Bits to choose from 2 to 5 membership functions, 2 Bits to choose one type of input function from Sigmoid, Bell Curve, Gaussian Curve and Two-sided Gaussian Curve and 1 Bit to choose the output function which can be Linear or Constant. In this way we end up with 21 bits chromosomes. The chromosome composition is illustrated in Figure 3.

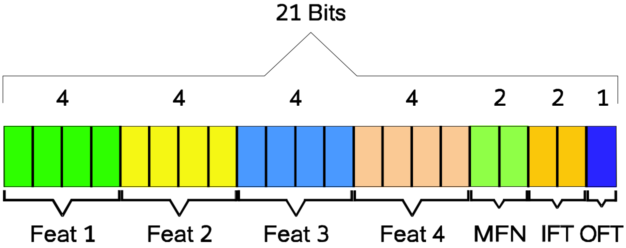
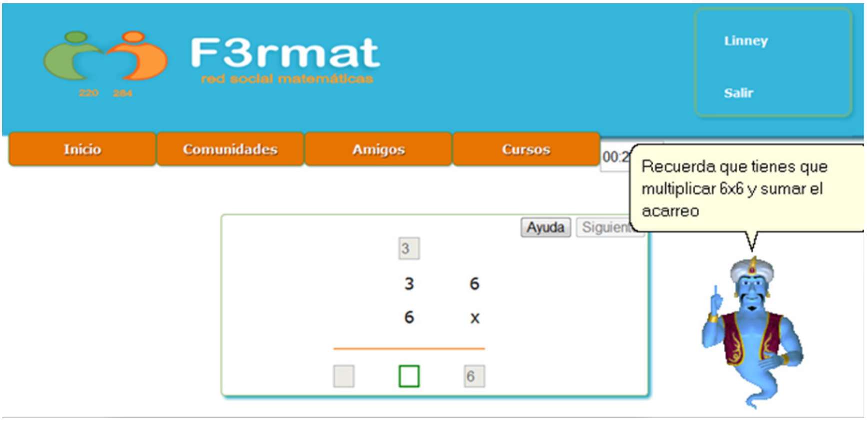


Fig. 3. Chromosome composition. Feat1-4 = Feature 1-4, MFN = Number of Membership Functions, IFT = Type of Input Function, OFT = Type of Output Function.

The final result from the genetic algorithm is the binary chromosome that reaches the best fitness score. In this case the chromosome was 0000 0100 0011 0110 01 10 0. This means that the best configuration was: 1st Feature: No feature was selected. 2nd Feature: Feature in position 4 which is Intensity Average. 3rd Feature: Feature in position 3 which is Pitch Range. 4th Feature: Feature in position 6 which is Shimmer. Number of Membership Functions = 5. Type of Input Functions = Bell Curve. And Type of Output Function = Constant.

Once the affective student state has been inferred from face and speech, both results are merged with the cognitive state and then sent to the affective module. To do that, the tutor needs a model that enables a mapping from the affective and knowledge student state to tutorial actions. The affective component of a tutorial action tries to promote a positive affective student state while the cognitive component tries to transmit knowledge. Figure 4 shows an interface of the ITS with an affective agent represented by the Genie character of Microsoft agent.

The actions of the Genie are shown in table 2. These actions were the results of studies conducted to evaluate the expressivity of the animated agents [22]. In these surveys, 20 teachers were asked to select appropriate affective actions to be presented according to several tutorial scenarios.



Recuerda que tienes que multiplicar 6x6 y sumar el acarreo

Fig. 4. Cognitive and affective feedback of the ITS

We want the agent's tutorial actions help students to learn and to foster a good affective state; hence we use decision theory to achieve the best balance between these two objectives. The decision process is represented as a dynamic decision network (DDN). Our model uses multi-attribute utility theory to define the necessary utilities [23]. That is, the DDN establishes the tutorial action considering two utility measures, one on learning and one on affect, which are combined to obtain the global utility by a weighted linear combination (figure 5). These utility functions are the means that allow educators adopting the system to express their preferences towards learning and affect.

Table 3. Animations preferred by the teachers

Affective Action
Acknowledge
Announce
Congratulate
Confused
Get Attention
Explain
Suggest
Think

After the student performs an action, i.e. after the student model is updated (time t_n), a new time slice is added (time t_{n+1}). At time t_n we have the current student state and the possible tutorial actions; at time t_{n+1} we have the prediction of how the tutor action influences the student's affect and knowledge, from which we estimate the individual and global utilities. The *affective state* in figure 5 is assessed by the affective module described before.

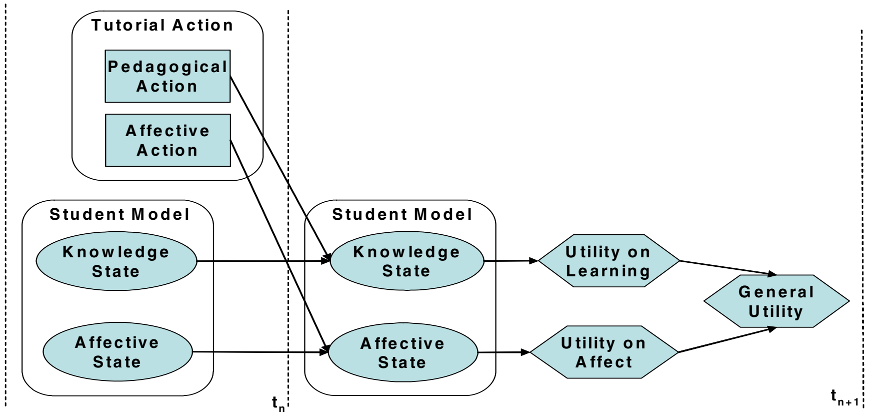


Fig. 5. The DDN to obtain a value from knowledge (cognitive) and affective states

Table 4. Results, Course Level, and Learning Method for 10 students

Student	Initial Grade	Final Grade	Improvement %	Course Level	Method
1	4.74	7.89	31.5	difficult	Traditional
2	3.68	4.21	5.3	normal	lattice
3	3.68	7.89	42.1	difficult	lattice
4	7.89	8.42	5.3	difficult	Traditional
5	7.37	8.42	10.5	difficult	Traditional
6	10.00	10.00	0	difficult	Traditional
7	6.84	8.42	15.8	difficult	Traditional
8	8.42	8.42	0	difficult	lattice
9	8.42	9.47	10.5	difficult	Traditional
10	3.68	4.21	5.3	normal	lattice
Total %	6.78	8.12	13.4		

4 Results and Discussions

The social network and the ITS were evaluated by a group of children from third grade (Figure 5). There were 72 children from public and private schools who tested the tool and the tutoring system. Before the evaluation we offered a small introduction to the tool. We evaluated the subject of multiplication. We applied a test before and after the students used the software tool. Table 3 shows the results of a random sample of 10 students. We can see from the results a good improvement in most students (more in students with lower initial grades) using one of the two teaching methods for multiplication: traditional and lattice.

Fig. 6. Children testing the social network and the ITS in public and private schools

5 Conclusions and Future Work

The social network along with the intelligent tutoring system was implemented using different software tools and programming languages. The presentation layer of the social network was implemented with CCS3, HTML 5 and Javascript. For the logical layer (ITS modules and social network functionalities) we use Java and JSP. For the data layer we use XML and MySQL.

We need to do more testing with more students. We still are working with the emotion recognizer (implement it in the social network) and we are adding more math operations to the ITS. We have implemented multiplications and divisions (elementary school). However, the initial results are encouraging. We are implementing our own recognizers because we need to use them in a Web platform where the social network and ITS may be accessed from any place and from any computer platform.

References

[1] Picard, R.W.: Affective Computing. MIT Technical Report #321 (1995)

[2] Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: Diminitrova, V., Mizoguchi, R., Du Boulay, B., Graesser, A. (eds.) Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 17–24. IOS Press, Amsterdam (2009)

[3] Conati, C., Maclare, H.: Evaluating a Probabilistic Model of Student Affect. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (eds.) ITS 2004. LNCS, vol. 3220, pp. 55–66. Springer, Heidelberg (2004)

[4] D'Mello, S.K., Picard, R.W., Graesser, A.C.: Towards an affective-sensitive AutoTutor. Special Issue on Inteligent Educational Systems. IEEE Intelligent Systems 22(4), 53–61 (2007)

[5] Carbonell, J.R.: AI in CAI: An artificial intelligence approach to computer-aided-instruction. IEEE Transactions on Man-Machine System, MMS 11(4), 190–202 (1970)

[6] Clancey, W.J.: Transfer of rule-based expertise through a tutorial dialogue. Computer Science Stanford, CA, Stanford University. PhD (1979)

[7] Anderson, R., Boyle, C.F., Corbett, A.T., Lewis, M.W.: Cognitive modeling and intelligent tutoring. Artificial Intelligence 42, 17–49 (1990), doi:10.1016/0004-3702(90)90093-F

[8] Aleven, V., Koedinger, K.: An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. Cognitive Science 26(2), 147–179 (2002)

[9] Woolf, B.P.: Building intelligent interactive tutors. Morgan Kaufmann (2009) ISBN: 978-0-12-373594-2

[10] D'Mello, S.K., Picard, R.W., Graesser, A.C.: Towards an affective-sensitive AutoTutor. Special Issue on Inteligent Educational Systems. IEEE Intelligent Systems 22(4), 53–61 (2007)

[11] Du Boulay, B.: Towards a motivationally intelligence pedagogy: how should an intelligent tutor respond to the unmotivated or the demotivated? New Perspectives on Affect and Learning Technologies, Explorations in the Learning Sciences, Instructional Systems and Performance Technologies 3, 41–52 (2011)

[12] Doignon, J.-P., Falmagne, J.C.: Knowledge Spaces. Springer (1999)

[13] Günel, K.: Intelligent Tutoring Systems: Conceptual Map Modeling. Lambert Academic Publishing (2010)

[14] Ekman, P.: Facial Expressions. John Wiley & Sons Ltd., New York (1999)

[15] Stathopoulou, I.O., Tsihrintzis, G.A.: Visual Affect Recognition. Frontiers in Artificial Intelligence and Applications, vol. 214. IOS Press (2010)

[16] Yang, M.H., Ahuja, N.: Face Detection and Gesture Recognition for Human-Computer Interaction. Kluwer Academic Publishers (2001)

[17] Sinha, P.: Object Recognition via image-invariants. Investigative Ophthalmology and Visual Science 35, 1735–1740 (1994)

[18] Tian, Y.-l., Kanade, T., Cohn, J.F.: Recognizing Action Units for Facial Expression Analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence (2001)

[19] Dellaert, F., Polizin, T., Waibel, A.: Recognizing Emotion in Speech. In: Proceeding. of ICSLP, Philadelphia, PA, pp. 1970–1973 (1996)

[20] Lee, C.M., Narayanan, S., Pieraccini, R.: Recognition of Negative Emotion in the Human Speech Signals. In: Workshop on Automatic Speech Recognition and Understanding (2001)

[21] Boersma, P., Weenink, D.: Praat v. 4.0.8, A system for doing phonetics by computer. Institute of Phonetic Sciences of the University of Amsterdam (2002)

[22] Hernández, Y., Sucar, L.E., Arroyo, G.: Building an affective model for intelligent tutoring systems with base on teachers' expertise, pp. 754–764. Springer (2008)

[23] Clemen, R.T.: Making hard decisions. Duxbury Press, Belmont (2000)

Towards a Personality Fuzzy Model Based on Big Five Patterns for Engineers Using an ANFIS Learning Approach

Luis G. Martínez[1], Juan R. Castro[1], Guillermo Licea[1], Antonio Rodríguez-Díaz[1], and Reynaldo Salas[2]

[1] Universidad Autónoma de Baja California
Calzada Universidad 14418, Mesa de Otay 22300 Tijuana B.C., México
{luisgmo,jrcastro,glicea,ardiaz }@uabc.edu.mx
[2] Tijuana Institute of Technology
Calzada Tecnológico, Frac. Tomás Aquino 22370 Tijuana B.C., México
reynaldo.salas@tectijuana.edu.mx

Abstract. This paper proposes an ANFIS (Adaptive Network Based Fuzzy Inference System) Learning Approach where we have found patterns of personality types using Big Five Personality Tests for students in Engineering Programs. An ANFIS model is applied to the personality traits of the Big Five Personality Model obtaining a Takagi-Sugeno-Kang (TSK) Fuzzy Inference System (FIS) type model with rules that are helping us identify Big Five Patterns for students that are studying Engineering Programs.

Keywords: Fuzzy Logic, Uncertainty, Big Five Personality Test.

1 Introduction

Many people going into adulthood have to decide to take up a career, job or profession. To make a career choice you have to base it on your skills, interests, values and traits, you can write them up and then analyze them to make a choice and plan a career, or take career tests to help evaluate yourself and discover your skills, interests, values and traits. Nowadays students are trying to find a better way to choose a career; from career tests, career planning, personality assessments, job advice and career counseling services, all an overwhelming load of information to choose from, especially online.

There are many different types of personality tests, commonly referred to as objective personality tests, involving many questions or items where test takers respond by rating the degree to which each item reflects their behavior scoring it objectively. A person's personality is the result of an interaction between the person and the environment, it can be defined as a dynamic and organized set of characteristics possessed by a person that uniquely influences his cognitions, emotions, motivations and behaviors in various situations, personality tests are design to evaluate these items.

I. Batyrshin and M. González Mendoza (Eds.): MICAI 2012, Part II, LNAI 7630, pp. 456–466, 2013.

The uses of personality tests in modern society include self reflection and understanding, job placement and for learning how to better interact with others in a team or work group. Effective use of psychometric instruments can add value and accuracy to the test. Many studies have found personality to be related to academic performance, choice of electives, completing university education, and choice of career [1][2][3]. This paper proposes an ANFIS (Adaptive Network Based Fuzzy Inference System) Learning Approach to find personality type patterns of engineers studying different programs, based on the Big Five Personality Test. Personality tests are based on interpretation; therefore to tackle uncertainty a Takagi-Sugeno-Kang (TSK) Fuzzy Inference System (FIS) type model with rules will help us identify better patterns for these engineers.

The rest of the paper is organized as follows: section 2 is a brief background of personality, careers in engineering and job performance relationships. Section 3 defines the methodology towards defining our Fuzzy Model. Section 4 displays results of the big five personality test, the big five patterns and implementation of ANFIS model, concluding in section 5 with observations and discussion of our case study.

2 Background

The engineering profession is one area of human endeavor in which there is a very high consistency in human character traits, we can say that an "engineering personality" exist [4]. The engineer's most obvious characteristics are his precision, his meticulousness, his attention to detail and accuracy, in other words his perfectionism.

Nagarjuna's [5] study comparing engineers with commerce students, shows that engineering students are more self-reliant, realistic, responsible and emotionally tough. Engineering students are more socially aware, controlled, self-disciplined and perfectionists as compared to the commerce background students. The curriculum for engineering course is more technical and focused and demands more perfection to be successful and also due to the fact that the engineering students spend more years with their classmates they tend to become tougher minded and focused as they are given lot of team assignments in their curriculum as compared to the commerce graduates. The students are imparted knowledge and skills that will help them in the decision making process and in efficient handling of day to day operations in a variety of professions. Factors like stress tolerance and team participation have become very essential in the current organizational context in achieving optimal results.

In 1980 a consortium of eight universities and the Center for Applications of Psychological Type was formed to study the role of personality type in engineering education. Introverts, intuitors, and judgers generally outperformed their extraverted, sensing, and perceiving counterparts in the population studied [6][7]. Wankat and Oreovicz [8] observe that if memorization and recall are important, sensing types should perform better, while if analysis is required, intuitive students should have an advantage. Rosati [9][10] also observed that introverts, thinkers, and judgers were more likely than extraverts, feelers, and perceivers to graduate in engineering after four years, but sensors were more likely than intuitors to do so.

Personality relationship with job performance has been a frequently studied topic in industrial psychology [11], it relates to how well employees perform their tasks, the initiative they take and the resourcefulness they show in solving problems. Barrick and Mount [12] and Salgado [13] found that Conscientiousness and Emotional Stability are valid predictors across job criteria, job performance and occupational groups, the remaining factors are valid only for some criteria and some occupational groups; also Openness and Agreeableness were valid predictors of training proficiency. Schneider [14] concluded that Extroversion and Conscientiousness predict job performance in various occupations. Relationships between personality dimensions and job performance and occupations are necessary, if these relationships are found accurately results can be used for recruitment, selection and career development purposes.

Also personality and career success has been studied [15]; it has been asserted that achievement can be explained largely by factors such as individual initiative, effort and merit, which are probably a significant determinant of how people will do in their careers. The dominant paradigm in literature on personality and job preferences comes from Holland's RIASEC (realistic-investigative-artistic-social-enterprising-conventional) model [16], it proposes that there are stable individual differences in preferences for job characteristics and that individuals who are in jobs that match their preferences will be more satisfied, the basic idea underlying this theory is 'Behavior is determined by an interaction between personality and environment'; it is assumed that your personality forms as a result of an interaction between you and your environment. As you develop a preference for certain activities and a dislike for others, interests crystallize, and you develop related skills and knowledge, you think and act in special ways, environments tend to be dominated by people with certain personalities who surround themselves with people like themselves and seek out problems that match their interests, talents and world outlook.

As we can see commonly held opinions about the relationship between some occupations and character traits they have considerable validity; some occupations do not show any consistent trend as far as personality traits are concerned, but for engineering careers we tend to see a pattern of traits, skills and therefore personality style. Additional inputs of these characteristics would give insight to counselors in job advising, career planning, and career counseling, shaping overall personality of the students so they can choose appropriately and wisely to fare better in the corporate world once graduated.

This paper specifically analyzes results of internet's free Big Five Test applied in a case study of students of engineering programs to find Big Five Patterns for these careers.

3 Methodology

At the University of Baja California, Tijuana Mexico we took a sample of 100 students from different engineering programs applying the Big Five Test to the group. Big Five personality tests claim to measure your intensities in relation to the "Big

Five" factors [17]. The structure of the tests requires selecting options from multiple choice questionnaires. These big five personality tests equate your personality to your collective degrees of behavior in five factors.

The Big Five factors are Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism (OCEAN, or CANOE if rearranged). The Neuroticism factor is sometimes referred to as Emotional Stability. And Openness factor sometimes is referred as Intellect.

Openness (O) is a disposition to be imaginative, inventive, curious, unconventional and autonomous, has an appreciation for art, emotion, adventure, unusual ideas, curiosity and variety of experience.

Conscientiousness (C) comprises of two related facets achievement and dependability, has a tendency to show self-discipline, be efficient, organized, act dutifully and aim for achievement, plans rather than behave spontaneously.

Extroversion (E) represents tendency to be sociable, outgoing and assertive, experiences positive affect such as energy, passion and excitement.

Agreeableness (A) is a tendency to be trusting, friendly, compassionate, cooperative, compliant, caring and gentle.

Neuroticism (N) represents tendency to exhibit poor emotional adjustment and experience negative or unpleasant emotions easily, such as anxiety, insecurity, depression and hostility, opposite of this trait is Emotional Stability (ES), as out big five test throws us a value of ES instead of N, we are considering ES as a linguistic variable and representing it with a label of N-1.

Because of uncertainty of personality traits a fuzzy based approach is considered to provide an integrated quantity measure for abilities of engineering careers which incorporates all aspects of personality traits involved in each engineering program. Recent studies are incorporating fuzzy approaches to personnel selection and job selection relating traits with careers [18][19][20].

The modeling approach used by ANFIS is similar to many system identification techniques. First, you hypothesize a parameterized model structure (relating inputs to membership functions to rules to outputs to membership functions, and so on). Next, you collect input/output data in a form that will be usable by ANFIS for training. You can then use ANFIS to train the FIS model to emulate the training data presented to it by modifying the membership function parameters according to a chosen error criterion. In general, this type of modeling works well if the training data presented to ANFIS for training (estimating) membership function parameters is fully representative of the features of the data that the trained FIS is intended to model.

This method has been applied to design intelligent systems for control [21][22], for pattern recognition, fingerprint matching and human facial expression recognition[23][24].

The ANFIS under consideration has five variable inputs denoted by $x = \{ O, C, E, A, N \}$, with two Gaussian membership functions (B), a set of 32 rules and one output variable Engineering Program (P). For a first-order Sugeno Fuzzy Model, a k-th rule can be expressed as:

Rule *k*-th:

IF (x_1 is B_1^k) AND (x_2 is B_2^k) AND (x_3 is B_3^k) AND (x_4 is B_4^k) AND (x_5 is B_5^k)
THEN R is $f^k(x)$,where

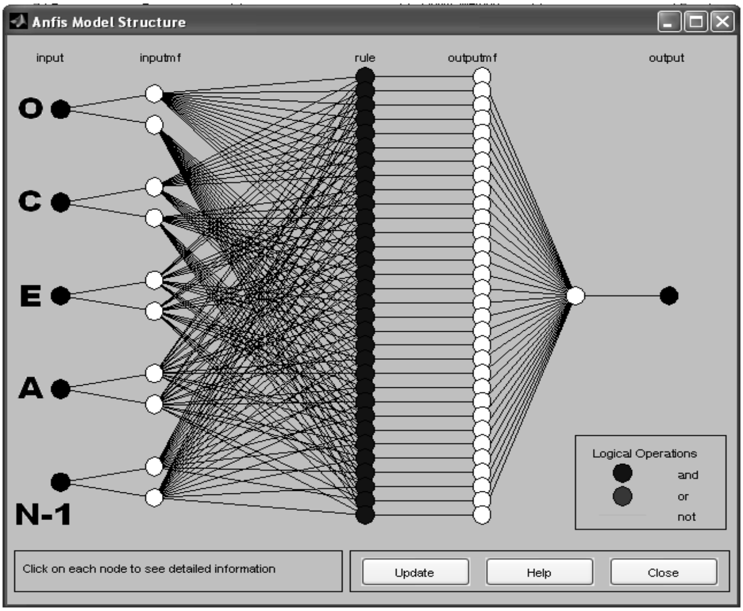
$$f^k(x) = p_1^k x_1 + p_2^k x_2 + p_3^k x_3 + p_4^k x_4 + p_5^k x_5 + p_0^k \quad . \tag{1}$$

and membership functions are denoted by:

$$\mu_{B_i^k}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - m_i^k}{\sigma_i^k}\right)\right] . \tag{2}$$

where p_i^k are linear parameters, and B_i^k are Gaussian membership functions for k = 1, 2, 3, …, 32 and i= 1, …, 5.

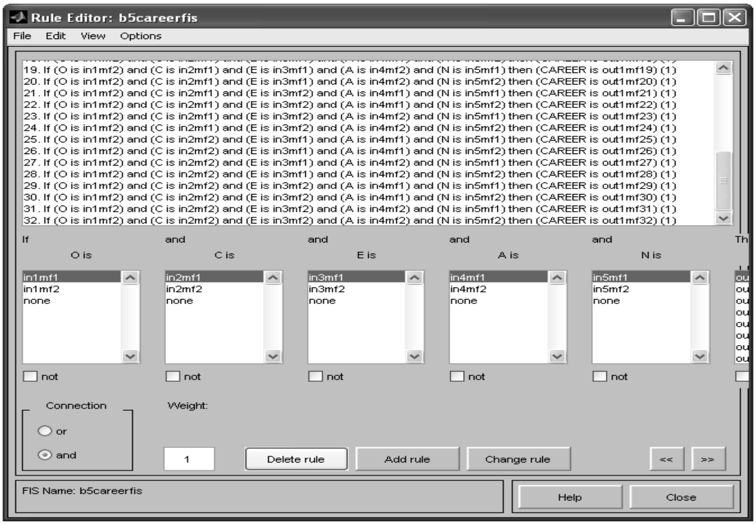
The corresponding equivalent ANFIS architecture is as shown in Fig. 1. The entire system architecture consists of five layers, these are { input, inputmf, rule, outputmf, output }.



Click on each node to see detailed information

Fig. 1. ANFIS Big Five Test Model Architecture

Using this architecture, with OCEAN traits as input linguistic variables and Engineering Programs as an output linguistic variable, a Takagi-Sugeno-Kang (TSK) FIS type model was developed using the ANFIS Editor GUI [25], creating, training and testing it, to find adequate rules to help us find the relationship of Engineering Programs with OCEAN personality traits. Figure 2 is an abstract visualization of the rules generated by the model.



```
19. If (O is in1mf2) and (C is in2mf1) and (E is in3mf1) and (A is in4mf2) and (N is in5mf1) then (CAREER is out1mf19) (1)
20. If (O is in1mf2) and (C is in2mf1) and (E is in3mf1) and (A is in4mf2) and (N is in5mf2) then (CAREER is out1mf20) (1)
21 . If (O is in1mf2) and (C is in2mf1) and (E is in3mf2) and (A is in4mf1) and (N is in5mf1) then (CAREER is out1mf21) (1)
22. If (O is in1mf2) and (C is in2mf1) and (E is in3mf2) and (A is in4mf1) and (N is in5mf2) then (CAREER is out1mf22) (1)
23. If (O is in1mf2) and (C is in2mf1) and (E is in3mf2) and (A is in4mf2) and (N is in5mf1) then (CAREER is out1mf23) (1)
24. If (O is in1mf2) and (C is in2mf1) and (E is in3mf2) and (A is in4mf2) and (N is in5mf2) then (CAREER is out1mf24) (1)
25. If (O is in1mf2) and (C is in2mf2) and (E is in3mf1) and (A is in4mf1) and (N is in5mf1) then (CAREER is out1mf25) (1)
26. If (O is in1mf2) and (C is in2mf2) and (E is in3mf1) and (A is in4mf1) and (N is in5mf2) then (CAREER is out1mf26) (1)
27. If (O is in1mf2) and (C is in2mf2) and (E is in3mf1) and (A is in4mf2) and (N is in5mf1) then (CAREER is out1mf27) (1)
28. If (O is in1mf2) and (C is in2mf2) and (E is in3mf1) and (A is in4mf2) and (N is in5mf2) then (CAREER is out1mf28) (1)
29. If (O is in1mf2) and (C is in2mf2) and (E is in3mf2) and (A is in4mf1) and (N is in5mf1) then (CAREER is out1mf29) (1)
30. If (O is in1mf2) and (C is in2mf2) and (E is in3mf2) and (A is in4mf1) and (N is in5mf2) then (CAREER is out1mf30) (1)
31 . If (O is in1mf2) and (C is in2mf2) and (E is in3mf2) and (A is in4mf2) and (N is in5mf1) then (CAREER is out1mf31) (1)
32. If (O is in1mf2) and (C is in2mf2) and (E is in3mf2) and (A is in4mf2) and (N is in5mf2) then (CAREER is out1mf32) (1)
```

FIS Name: b5careerfis

Fig. 2. Rules obtained in ANFIS Big Five Test Model

4 Results

Work of our case study consisted of 100 engineering students distributed as follows: 12 aerospace engineers (AE), 10 bioengineers (BI), 30 computer engineers (CE), 10 chemical engineers (CH), 17 electronic engineers (EE) and 21 industrial engineers (IE).

Big Five personality test results obtained are presented in Table 1 showing the means and standard deviation of each trait related for every engineering program (EP). Each table row is a personality vector unique for a student of each EP in question, no two rows have exactly the same values for every one of their attributes, and this gives us a significant difference between each program student to distinguish patterns based on the Big Five Personality Test.

There are two ways to evaluate this data; first we can analyze the Trait Personality Vector Ev for extroversion trait, with values Ev = {43.33, 58.8, 44.27, 50.4, 38.47, 52.33}, meaning that a high score on this trait by the student it is possible to recommend that person to be a Bioengineer, if it’s a low score then we can recommend this person to study electronic engineer. A second way to analyze is to consider the Engineering Program Vector, for example EEv vector with values EEv = {51.18, 59.65, 38.47, 47.18, 56.35}, means that an EE program can be recommended for a person with high scores in (C) Conscientiousness and a low score of (E) Extroversion.

To see a broader picture these results can be displayed relating them with a center point using a radar chart type, obtaining Big Five Patterns (B5P) for Engineering Programs, these patterns are shown in figure 3, while figure 4 is a comparative of them.

Table 1. Results of OCEAN test for Engineering Programs.

	O	C	E	A	N-1
AE	64.17	51	43.33	55.17	58
BI	54.6	46.6	58.8	45.5	55
CE	50.87	58.93	44.27	48.93	51.63
CH	53.6	62	50.4	46.6	54.6
EE	51.18	59.65	38.47	47.18	56.35
IE	44.38	56.76	52.33	45.14	59.52
	STANDARD DEVIATION				
AE	9.003	10.57	14.1	8.548	10.3
BI	1.647	13.43	7.131	6.311	7.789
IC	9.198	10.71	12.07	8.658	13.04
CH	10.1	8.219	10.57	7.604	10.79
EE	10.49	10.23	12.58	11.14	10.66
IE	12.55	12.77	9.702	8.662	13.4

AE = aerospace engineers, BI= bioengineers, CE= computer engineers, CH= chemical engineers, EE= electronic engineers, IE= industrial engineers. O=Openness, C=Conscientiousness, E=Extroversion, A=Agreeableness, N-1=Emotional Stability

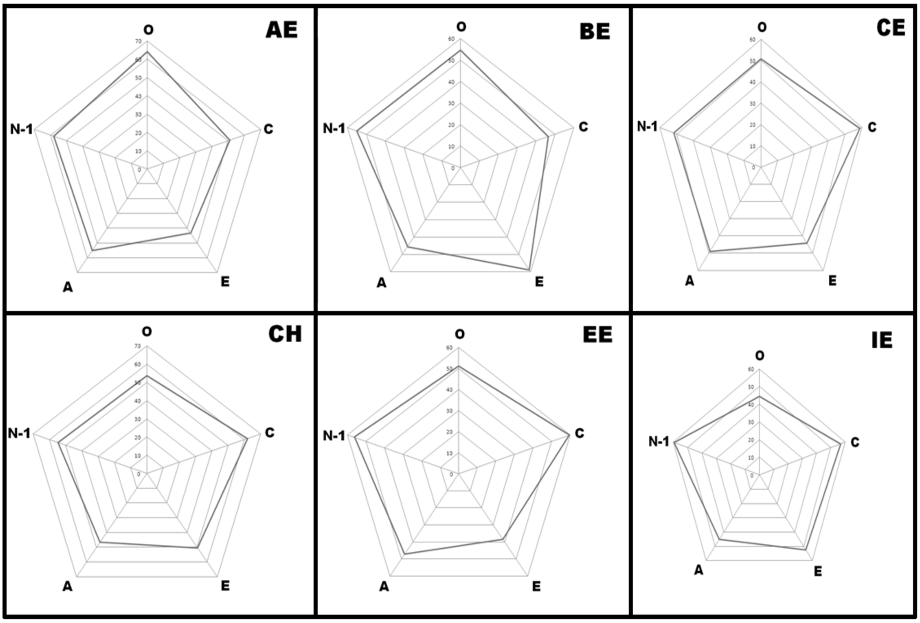


Fig. 3. Engineering Programs Big Five Patterns (B5P)

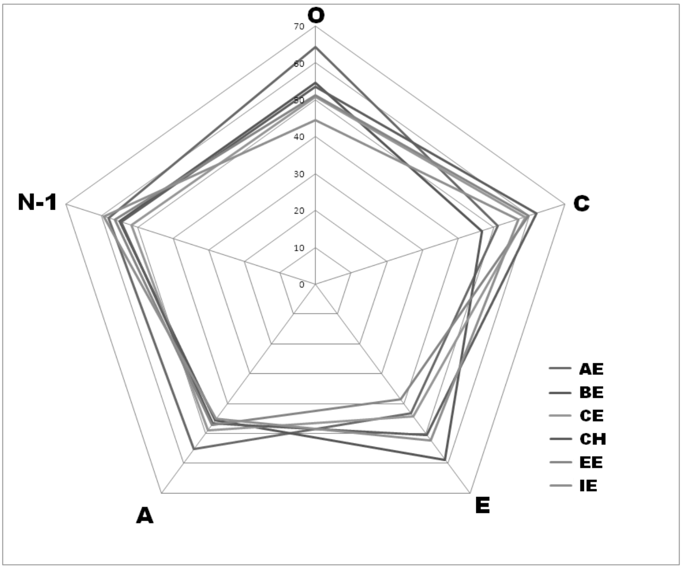


Fig. 4. Big Five Patterns Comparative

There are significant differences between each type of student. For example an Aerospace Engineer has highest values for traits (O) and (A) compared to an Industrial Engineer that has lowest values for these same traits. This comparison can give us a glimpse of specific traits for particular type of engineers, as always we should know that one trait does not define the personality of a type of engineer, but a personality vector with all traits involved can give us differences between types of student engineers.

These data was used for our ANFIS model obtaining figure 5 where it shows Input Trait and Output Engineering Program (EP) Relationships of this model. Whereas numeric values for linguistic variable EP were (1) Aerospace Engineer, (2) Bioengineer, (3) Computer Engineer, (4) Chemical Engineer, (5) Electonic Engineer and (6) Industrial Engineer.

The Fuzzy Model relates each trait with each EP, analizing data, range of trait means are from 40 to 60, we will consider low degree around 20-40 and high degree around 60-80 based on standard deviation. With these results and ranges we can assertain that a high degree of trait values for (C) and (N-1) we are talking about EP students of AE, BI or CE, they are emotionally stable, efficient, organized, aim for achievement. For a low degree of trait values (O) and (A) we are talking about EP students of II and EE, they are less creative, less trustworthy, less cooperative.

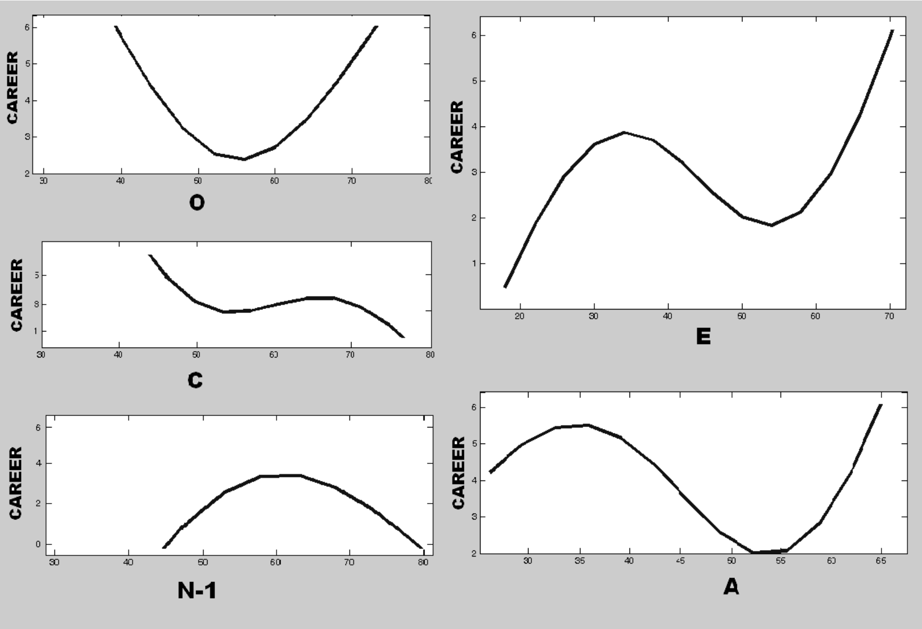


Fig. 5. Input Trait and Output Role Relationships

Comparing the B5P highest values we can observe that AE students have (O) and (A) very high, for emotional stability II registered highest value, for (E) are BI, for (C) are CH students; revising low values BI have low (C) trait, II have low (O), IE have low (E) meaning IE are highly introverted.

In psychology locus of control (LC) is a personality theory referring to the extent to which individuals believe they can control events that affect them. Locus of control describes the degree to which individuals perceive the outcome results from their own behaviors or from forces that are external to themselves [26]. Individuals with high internal LC believe that events in their life derive from their own actions, have better control of their own behavior, tend to be more politically involved and more likely to influence others, Work hard to develop their knowledge, skills and abilities; individuals with high external LC blame events on other people, not them, are worried about how other people see them so they are considerate and easy-going.

With this in mind the B5P for AE students have a high external LC as (O) and (A) traits are high, as they are creative and very cooperative, they depend of others opinions, how they see them, so they tend to get along and are very compliant. Opposite to II having a high internal LC as same traits are low and (N-1) and (E) traits are relatively high, they are more stable, more in control, their extroversion gives them an edge to influence others.

5 Conclusions

Our case study results have shed insight on a pattern of personality traits in relationship with engineering careers, there are significant differences between each EP vector, which is why we can only analyze our results viewing a student/professional profile in its totality. Further study of these patterns and individual traits can help us identify the personality of people working in engineering areas, as students and also as professionals.

First of all the utility of our study is directed to help counselors in job advising, career planning and career counseling, our ANFIS model can be a tool to help counselors validate student's decision to choose a career appropriately.

Also in curricula design identifying the type of students an EP has, can help us design the type of courses, learning methodology and the best teaching-learning approach to use; these would impact on skill and competency acquisition of students and make them more adaptive to their work and knowledge cognition.

In the professional world in job management our model can help us identify the motivation we could use for the type of engineer we are hiring to do the job, as identifying the LC will help us know if they are easy-going and depend on themselves or need a realistic sense of their circle of influence in order to experience success, therefore depending of others.

Future work in our fuzzy model has to better define the relationship between personality traits and EP in relation with the LC, as the uncertainty of LC has to be better validated. Research summarized in this paper and research that remains to be done will help students, professionals and counselors to orient people and better situate themselves for a better job occupation and career success.

References

1. Tett, R.P., Jackson, D.M., Rothstein, M.: Personality measures as predictors of job performance; a meta analytic review. Personnel Psychology 44, 703–742 (1991)
2. Singh, S.: Gender differences in work values and personality characteristics among Indian executives. Journal of Social Psychology 134, 699–700 (1994)
3. Song, Z., Wu, Q., Zhao, F.: Academic achievement and personality of college students. Chinese Mental Health Journal 16, 121–123 (2002)
4. Goshen, C.E.: The Engineer Personality. The Bent of Tau Beta Pi. The Engineering Honor Society, 15–20 (1954)
5. Nagarjuna, V.L., Mamidenna, S.: Personality Characteristics of Commerce and Engineering Graduates – A Comparative Study. Journal of the Indian Academy of Applied Psychology 34(2), 303–308 (2008)
6. McCaulley, M.H., Godleski, E.S., Yokomoto, C.F., Harrisberger, L., Sloan, E.D.: Applications of Psychological Type in Engineering Education. Engineering Education 73, 394–400 (1983)
7. McCaulley, M.H., Macdaid, G.P., Granade, J.G.: ASEE-MBTI Engineering Consortium: Report of the First Five Years. In: Proceedings,1985 ASEE Annual Conference, ASEE North Carolina (1985)
8. Wankat, P., Oreovicz, F.: Teaching Engineering. McGraw-Hill, New York (1993)

9. Rosati, P.: Student Retention from First-Year Engineering Related to Personality Type. In: Proceedings Frontiers in Education Conference, Washington DC, pp. 37–39. IEEE (1993)
10. Rosati, P.: A Personality Type Perspective of Canadian Engineering Students. In: Proceedings, Annual Conference of the Canadian Society of Civil Engineering, CSCE, vol. 4, pp. 11–20 (1999)
11. Rothmann, S., Coetzer, E.P.: The Big Five Personality Dimensions and Job Performance. Journal of Industrial Psychology 29(1), 68–74 (2003)
12. Barrick, M.R., Mount, M.K.: The big five personality dimensions and job performance: A meta-analysis. Personnel Psychology 44, 1–26 (1991)
13. Salgado, J.F.: The five-factor model of personality and job performance in the European Community. Journal of Applied Psychology 82, 30–43 (1997)
14. Schneider, M.H.: The relationship of personality and job settings to job satisfaction. Dissertation Abstracts International: Section B: Science and Engineering 59, 6103 (1999)
15. Judge, T.A., Kammeyer-Mueller, J.D.: Personality and Career Success. In: Handbook of Career Studies, ch. 4, pp. 57–78. Sage Publications (2007)
16. Holland, J.L.: Making vocational choices: A theory of vocational personalities and work environments. Prentice Hall, Odessa (1997)
17. Costa Jr., P.T., McCrae, R.R.: Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI) manual. Psychological Assessment Resources, Odessa (1992)
18. Selcuk, H., Cevikcan, E.: Job selection based on fuzzy AHP: An investigation including the students of Istanbul Technical University Management Faculty. Int. Journal of Business and Management Studies 3(1), 173–182 (2011)
19. Dereli, T., Durmusoglu, A., Ulusam, S.S., Avlanmaz, N.: A fuzzy approach for personnel selection process. TJFS: Turkish Journal of Fuzzy Systems 1(2), 126–140 (2010)
20. Nobari, S., Jabrailova, Z., Nobari, A.: Design a Fuzzy Model for Decision Support Systems in the Selection and Recruitment. In: Int. Conference on Innovation and Information Management. IPCSIT, vol. 36, pp. 195–200. IACSIT Press, Singapore (2012)
21. Aguilar, L., Melin, P., Castillo, O.: Intelligent control of a stepping motor drive using a hybrid neuro-fuzzy ANFIS approach. Applied Soft Computing 3(3), 209–219 (2003)
22. Melin, P., Castillo, O.: Intelligent control of a stepping motor drive using an adaptive neuro-fuzzy inference system. Inf. Sci. 170(2-4), 133–151 (2005)
23. Hui, H., Song, F.-J., Widjaja, J., Li, J.-H.: ANFIS-based fingerprint matching algorithm. Optical Engineering 43 (2004)
24. Gomathi, V., Ramar, K., Jeeyakumar, A.S.: Human Facial Expression Recognition Using MANFIS Model. Int. J. of Computer Science and Engineering 3(6), 335–339 (2009)
25. Fuzzy Logic Toolbox: User's Guide of Matlab. The Mathworks, Inc. (1995-2009)
26. Rotter, J.B.: Generalized expectancies for internal versus external control of reinforcement. Psychological Monographs 80(1), 1–28 (1966)

Author Index