LNAI 7095

Ildar Batyrshin
Grigori Sidorov (Eds.)

Advances in Soft Computing

10th Mexican International Conference
on Artificial Intelligence, MICAI 2011
Puebla, Mexico, November/December 2011, Proceedings, Part II

2 Part II

MICAI 2011

Springer

Lecture Notes in Artificial Intelligence 7095

Subseries of Lecture Notes in Computer Science

Ildar Batyrshin Grigori Sidorov (Eds.)

Advances in Soft Computing

10th Mexican International Conference
on Artificial Intelligence, MICAI 2011
Puebla, Mexico, November 26 – December 4, 2011
Proceedings, Part II

Springer

Volume Editors

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

Grigori Sidorov
National Polytechnic Institute (IPN)
Center for Computing Research (CIC)
Av. Juan Dios Bátiz, s/n, Col. Nueva Industrial Vallejo
Mexico D.F., CP 07738, Mexico
E-mail: sidorov@cic.ipn.mx

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-25329-4 e-ISBN 978-3-642-25330-0
DOI 10.1007/978-3-642-25330-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011940855

CR Subject Classification (1998): I.2, I.2.9, I.4, F.1, I.5.4, H.3-4

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 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.

This year's event was very special: we celebrated the 25th anniversary of SMIA and 10th anniversary edition of the MICAI series.

MICAI conferences traditionally publish high-quality papers in all areas of artificial intelligence 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 and 6438. Since its foundation in 2000, the conference has been growing in popularity and improving in quality.

The proceedings of MICAI 2011 have been published in two volumes. The first volume, *Advances in Artificial Intelligence*, contains 50 papers structured into five sections:

- Automated Reasoning and Multi-agent Systems
- Problem Solving and Machine Learning
- Natural Language Processing
- Robotics, Planning and Scheduling
- Medical Applications of Artificial Intelligence

The second volume, *Advances in Soft Computing*, contains 46 papers structured into five sections:

- Fuzzy Logic, Uncertainty and Probabilistic Reasoning
- Evolutionary Algorithms and Other Naturally Inspired Algorithms
- Data Mining
- Neural Networks and Hybrid Intelligent Systems
- Computer Vision and Image Processing

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 348 papers submitted for evaluation, by 803 authors from 40 countries; of these, 96 papers were selected for publication after a peer-reviewing process carried out by the international Program Committee. The acceptance rate was 27.5%.

The distribution of submissions by country or region is represented in Fig. 1, where the square of each circle corresponds to the number of submitted papers. Table 1 shows more detailed statistics. In this table, the number of papers is by authors: e.g., for a paper by 2 authors from USA and 1 author from UK, we added 2/3 to USA and 1/3 to UK.

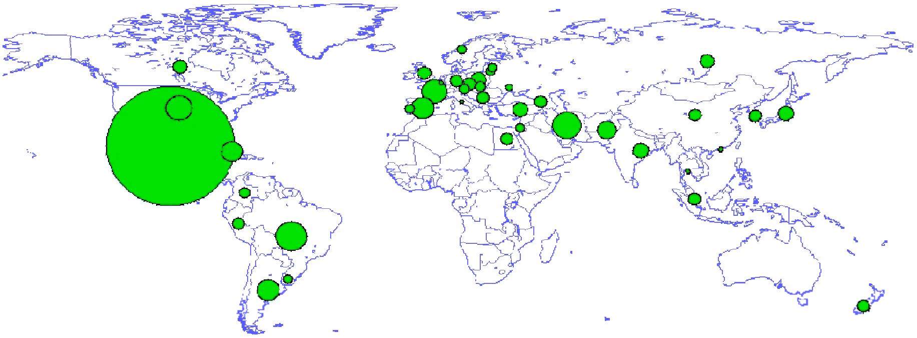


Fig. 1. Distribution of submissions by country or region.

Table 1. Submitted and accepted papers by country or region.

Country or region	Authors	Subm.	Acc.	Country or region	Authors	Subm.	Acc.
Argentina	13	7	3	Latvia	1	1	1
Austria	3	1.53	0.33	Lithuania	9	1	—
Belgium	1	0.25	—	Mexico	527	227.64	62.27
Brazil	35	13.25	3	New Zealand	5	2	1
Canada	8	2.6	1.6	Norway	1	1	—
China	5	2	—	Pakistan	11	4.92	1.42
Colombia	3	1.5	0.5	Peru	3	2	1
Cuba	15	6.21	1.75	Poland	5	3	1
Czech Rep.	4	2.5	1	Portugal	4	1	—
Egypt	5	2	—	Russian Federation	7	2.67	1
France	25	8.95	3.12	Serbia	4	2	—
Georgia	2	2	—	Singapore	2	2	1
Germany	3	2	1	Slovakia	2	1.5	—
Hong Kong	1	0.33	0.33	Spain	24	7.07	2.42
India	8	3.42	0.75	Thailand	1	0.33	—
Iran	16	11	2	Turkey	4	3	—
Israel	3	1.17	0.67	Ukraine	2	0.5	0.5
Italy	1	0.17	—	United Kingdom	6	2.32	1.32
Japan	7	3	1	United States	19	9.18	3.03
Korea, Rep. of	5	2	—	Uruguay	3	1	—

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: *SC Spectra: A New Soft Cardinality Approximation for Text Comparison*, by Sergio Jimenez Vargas and Alexander Gelbukh (Colombia, Mexico)
2nd place: *Fuzzified Tree Search in Teal Domain Games*, by Dmitrijs Rutko (Latvia)
3rd place: *Multiple Target Tracking with Motion Priors*, by Francisco Madrigal, Jean-Bernard Hayet and Mariano Rivera (Mexico)

In addition, the authors of the following papers selected among articles where the first author was a full-time student (excluding the papers listed above) received the Best Student Paper Award:

1st place: *Topic Mining Based on Graph Local Clustering*, by Sara Elena Garza Villarreal and Ramon Brena (Mexico)
2nd place: *Learning Probabilistic Description Logics: A Framework and Algorithms*, by Jose Eduardo Ochoa-Luna, Kate Revoredo and Fabio Gagliardi Cozman (Brazil)
3rd place: *Instance Selection Based on the Silhouette Coefficient Measure for Text Classification*, by Debangana Dey, Thamar Solorio, Manuel Montes y Gomez and Hugo Jair Escalante (USA, 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 Benemérita Universidad Autónoma de Puebla (BUAP), the Rector's Office of the BUAP headed by Dr. Enrique Agüera Ibañez; Dr. José Ramón Eguibar Cuenca, Secretary General of the BUAP; Alfonso Esparza Ortiz, Treasurer General of the BUAP; José Manuel Alonso of DDIE; Damián Hernández Méndez of DAGU; Dr. Lilia Cedillo Ramírez, Vice-rector of Extension and Dissemination of Culture of the BUAP; Dr. Gabriel Pérez Galmichi of the Convention Center; Dr. Roberto Contreras Juárez, Administrative Secretary of the Faculty of Computer Science of the BUAP; and to MC Marcos González Flores, head of the Faculty of Computer Science of the BUAP, for their warm hospitality related to MICAI 2011 and for providing the infrastructure for the keynote talks, tutorials and workshops, as well as for their valuable participation and support in the organization of this conference.

Their commitment allowed the opening ceremony, technical talks, workshops and tutorials to be held at the Centro Cultural Universitario, an impressive complex of buildings that bring together expressions of art, culture and academic affairs associated with the BUAP.

We are deeply grateful to the conference staff and to all members of the Local Committee headed by Dr. David Eduardo Pinto Avendaño. In particular, we would like to thank Dr. Maya Carrillo for chairing the logistic affairs of the conference, including her valuable effort for organizing the cultural program; Dr. Lourdes Sandoval for heading the promotion staff; as well as Dr. Arturo Olvera, head of the registration staff, Dr. Iván Olmos, Dr. Mario Anzures, and Dr. Fernando Zacarías (sponsors staff) for obtaining additional funds for this conference.

We also want to thank the sponsors that provided partial financial support to the conference: CONCYTEP, INAOE, Consejo Nacional de Ciencia y Tecnología (CONACYT) project 106625, TELMEX, TELCEL, Universidad Politécnica de

Puebla, UNIPUEBLA and Universidad del Valle de Puebla. We also thank Consejo de Ciencia y Tecnología del Estado de Hidalgo for partial financial support through the project FOMIX 2008/97071. We 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-DST (India) project “Answer Validation through Textual Entailment.”

The entire submission, reviewing and selection process as well as putting together the proceedings were 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 preparation of this volume.

September 2011

Ildar Batyrshin
Grigori Sidorov

Conference Organization

MICAI 2011 was organized by the Mexican Society of Artificial Intelligence (SMIA, Sociedad Mexicana de Inteligencia Artificial) in collaboration with Benemérita Universidad Autónoma de Puebla (BUAP), 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), Mexico.

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

Conference Committee

General Chair	Raúl Monroy
Program Chairs	Ildar Batyrshin and Grigori Sidorov
Workshop Chair	Alexander Gelbukh
Tutorials Chairs	Felix Castro Espinoza and Sofía Galicia Haro
Keynote Talks Chair	Jesus A. Gonzalez
Financial Chair	Grigori Sidorov
Grant Chairs	Raúl Monroy, Grigori Sidorov and Ildar Batyrshin
Best Thesis Awards Chair	Miguel Gonzalez
Doctoral Consortium Chairs	Oscar Herrera and Miguel Gonzalez
Organizing Committee Chair	David Pinto Avendaño

Track Chairs

Natural Language Processing	Sofia Galicia Haro
Machine Learning and Pattern Recognition	Mario Koeppen
Hybrid Intelligent Systems and Neural Networks	Sergio Ledesma Orozco
Logic, Reasoning, Ontologies, Knowledge Management, Knowledge-Based Systems, Multi-agent Systems and Distributed AI	Miguel González and Raul Monroy
Data Mining	Felix Castro Espinoza
Intelligent Tutoring Systems	Alexander Gelbukh
Evolutionary Algorithms and Other Naturally Inspired Algorithms	Nareli Cruz Cortés
Computer Vision and Image Processing	Oscar Herrera
Fuzzy Logic, Uncertainty and Probabilistic Reasoning	Alexander Tulupyev
Bioinformatics and Medical Applications	Jesús A. González
Robotics, Planning and Scheduling	Fernando Montes

Program Committee

Carlos Acosta
Hector-Gabriel Acosta-Mesa
Luis Aguilar
Ruth Aguilar
Esma Aimeur
Teresa Alarcón
Alfonso Alba
Rafik Aliev
Adel Alimi
Leopoldo Altamirano
Matias Alvarado
Gustavo Arechavaleta
Gustavo Arroyo
Serge Autexier
Juan Gabriel Aviña Cervantes
Victor Ayala-Ramirez
Andrew Bagdanov
Javier Bajo
Helen Balinsky
Sivaji Bandyopadhyay
Maria Lucia Barrón-Estrada
Roman Barták
Ildar Batyrshin (Chair)
Salem Benferhat
Tibebe Beshah
Albert Bifet
Igor Bolshakov
Bert Bredeweg
Ramon Brena
Paul Brna
Peter Brusilovsky
Pedro Cabalar
Abdiel Emilio Caceres Gonzalez
Felix Calderon
Nicoletta Calzolari
Gustavo Carneiro
Jesus Ariel Carrasco-Ochoa
Andre Carvalho
Mario Castelán
Oscar Castillo
Juan Castro
Félix Agustín Castro Espinoza
Gustavo Cerda Villafana
Mario Chacon
Lee Chang-Yong
Niladri Chatterjee
Zhe Chen
Carlos Coello
Ulises Cortes
Stefania Costantini
Raúl Cruz-Barbosa
Nareli Cruz-Cortés
Nicandro Cruz-Ramirez
Oscar Dalmau
Ashraf Darwish
Justin Dauwels
Radu-Codrut David
Jorge De La Calleja
Carlos Delgado-Mata
Louise Dennis
Bernabe Dorronsoro
Benedict Du Boulay
Hector Duran-Limon
Beatrice Duval
Asif Ekbal
Boris Escalante Ramírez
Jorge Escamilla Ambrosio
Susana C. Esquivel
Claudia Esteves
Julio Cesar Estrada Rico
Gibran Etcheverry
Eugene C. Ezin
Jesus Favela
Claudia Feregrino
Robert Fisher
Juan J. Flores
Claude Frasson
Juan Frausto-Solis
Olac Fuentes
Sofia Galicia-Haro
Ma.de Guadalupe Garcia-Hernandez
Eduardo Garea
Leonardo Garrido
Alexander Gelbukh
Onofrio Gigliotta
Duncan Gillies

Fernando Gomez
Pilar Gomez-Gil
Eduardo Gomez-Ramirez
Felix Gonzales
Jesus Gonzales
Arturo Gonzalez
Jesus A. Gonzalez
Miguel Gonzalez
José-Joel Gonzalez-Barbosa
Miguel Gonzalez-Mendoza
Felix F. Gonzalez-Navarro
Rafael Guzman Cabrera
Hartmut Haehnel
Jin-Kao Hao
Yasunari Harada
Pitoyo Hartono
Rogelio Hasimoto
Jean-Bernard Hayet
Donato Hernandez Fusilier
Oscar Herrera
Ignacio Herrera Aguilar
Joel Huegel
Michael Huhns
Dieter Hutter
Pablo H. Ibarguengoytia
Mario Alberto Ibarra-Manzano
Héctor Jiménez Salazar
Moa Johansson
W. Lewis Johnson
Leo Joskowicz
Chia-Feng Juang
Hiroharu Kawanaka
Shubhalaxmi Kher
Ryszard Klempous
Mario Koeppen
Vladik Kreinovich
Sergei Kuznetsov
Jean-Marc Labat
Susanne Lajoie
Ricardo Landa Becerra
H. Chad Lane
Reinhard Langmann
Bruno Lara
Yulia Ledeneva
Ronald Leder
Sergio Ledesma-Orozco
Yoel Ledo Mezquita
Eugene Levner
Derong Liu
Weiru Liu
Giovanni Lizarraga
Aurelio Lopez
Omar Lopez
Virgilio Lopez
Gabriel Luque
Sriram Madurai
Tanja Magoc
Luis Ernesto Mancilla
Claudia Manfredi
J. Raymundo Marcial-Romero
Antonio Marin Hernandez
Luis Felipe Marin Urias
Urszula Markowska-Kaczmar
Ricardo Martinez
Edgar Martinez-Garcia
Jerzy Martyna
Oscar Mayora
Gordon Mccalla
Patricia Melin
Luis Mena
Carlos Merida-Campos
Efrén Mezura-Montes
Gabriela Minetti
Tanja Mitrovic
Dieter Mitsche
Maria-Carolina Monard
Luís Moniz Pereira
Raul Monroy
Fernando Martin Montes-Gonzalez
Manuel Montes-y-Gómez
Oscar Montiel
Jaime Mora-Vargas
Eduardo Morales
Guillermo Morales-Luna
Enrique Munoz de Cote
Angel E. Munoz Zavala
Angelica Munoz-Melendez
Masaki Murata
Rafael Murrieta
Tomoharu Nakashima

Additional Reviewers

Aboura, Khalid
Acosta-Guadarrama, Juan-Carlos
Aguilar Leal, Omar Alejandro
Aguilar, Ruth
Arce-Santana, Edgar
Bankevich, Anton
Baroni, Pietro
Bhaskar, Pinaki
Bolshakov, Igor
Braga, Igor
Cerda-Villafana, Gustavo
Chaczko, Zenon
Chakraborty, Susmita
Chavez-Echeagaray, Maria-Elena
Cintra, Marcos
Confalonieri, Roberto
Darriba, Victor
Das, Amitava
Das, Dipankar
Diaz, Elva
Ezin, Eugene C.
Figueroa, Ivan
Fitch, Robert
Flores, Marisol
Gallardo-Hernández, Ana Gabriela
Garcia, Ariel
Giacomin, Massimiliano
Ibarra Esquer, Jorge Eduardo
Joskowicz, Leo
Juárez, Antonio
Kawanaka, Hiroharu
Kolesnikova, Olga
Ledeneva, Yulia
Li, Hongliang
Lopez-Juarez, Ismael
Montes Gonzalez, Fernando
Murrieta, Rafael
Navarro-Perez, Juan-Antonio
Nikodem, Jan
Nurk, Sergey
Ochoa, Carlos Alberto
Orozco, Eber
Pakray, Partha
Pele, Ofir
Peynot, Thierry
Piccoli, María Fabiana
Ponomareva, Natalia
Pontelli, Enrico
Ribadas Pena, Francisco Jose
Rodriguez Vazquez, Katya
Sánchez López, Abraham
Sirotkin, Alexander
Suárez-Araujo, Carmen Paz
Villatoro-Tello, Esaú
Wang, Ding
Yaniv, Ziv
Zepeda, Claudia

Organizing Committee

Local Chair	David Pinto Avendaño
Logistics Staff	Maya Carrillo
Promotion Staff	Lourdes Sandoval
Sponsors Staff	Ivan Olmos, Mario Anzures, Fernando Zacarías
Administrative Staff	Marcos González and Roberto Contreras
Registration Staff	Arturo Olvera

Table of Contents – Part II

Fuzzy Logic, Uncertainty and Probabilistic Reasoning

Evolutionary Algorithms and Other Naturally-Inspired Algorithms

Data Mining

Neural Networks and Hybrid Intelligent Systems

Computer Vision and Image Processing

Table of Contents – Part I

Automated Reasoning and Multi-Agent Systems

Problem Solving and Machine Learning

Natural Language Processing

Robotics, Planning and Scheduling

Medical Applications of Artificial Intelligence

Intelligent Control of Nonlinear Dynamic Plants Using a Hierarchical Modular Approach and Type-2 Fuzzy Logic

Leticia Cervantes, Oscar Castillo, and Patricia Melin

Tijuana Institute of Technology
ocastillo@tectijuana.mx

Abstract. In this paper we present simulation results that we have at this moment with a new approach for intelligent control of non-linear dynamical plants. First we present the proposed approach for intelligent control using a hierarchical modular architecture with type-2 fuzzy logic used for combining the outputs of the modules. Then, the approach is illustrated with two cases: aircraft control and shower control and in each problem we explain its behavior. Simulation results of the two case show that proposed approach has potential in solving complex control problems.

Keywords: Granular computing, Type-2 fuzzy logic, Fuzzy control, Genetic Algorithm.

1 Introduction

This paper focuses on the field of fuzzy logic, granular computing and also considering the control area. These areas can work together to solve various control problems, the idea is that this combination of areas would enable even more complex problem solving and better results. We explain and illustrate the proposed approach with some control problems, one is the automatic design of fuzzy systems for the longitudinal control of an airplane using genetic algorithms. This control is carried out by controlling only the elevators of the airplane. To carry out such control it is necessary to use the stick, the rate of elevation and the angle of attack. These 3 variables are the inputs to the fuzzy inference system, which is of Mamdani type, and we obtain as output the values of the elevators. For optimizing the fuzzy logic control design we use a genetic algorithm. We also illustrate the approach of fuzzy control with the benchmark case of shower control. Simulation results show the feasibility of the proposed approach of using hierarchical genetic algorithms for designing type-2 fuzzy systems.

The rest of the paper is organized as follows: In section 2 we present some basic concepts to understand this work, in section 3 we define the proposed method, section 4 describes automatic design of a fuzzy system for control of aircraft dynamic system genetic optimization, Section 5 presents a hierarchical genetic algorithm for optimal type-2 fuzzy system design, and finally conclusions are presented in section 6.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 1–12, 2011.

2 Background and Basic Concepts

We provide in this section some basic concepts needed for this work.

2.1 Granular Computing

Granular computing is based on fuzzy logic. There are many misconceptions about fuzzy logic. To begin with, fuzzy logic is not fuzzy. Basically, fuzzy logic is a precise logic of imprecision. Fuzzy logic is inspired by two remarkable human capabilities. First, the capability to reason and make decisions in an environment of imprecision, uncertainty, incompleteness of information, and partiality of truth. And second, the capability to perform a wide variety of physical and mental tasks based on perceptions, without any measurements and any computations. 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[2][6][30][39][40].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[3][4][6].

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.

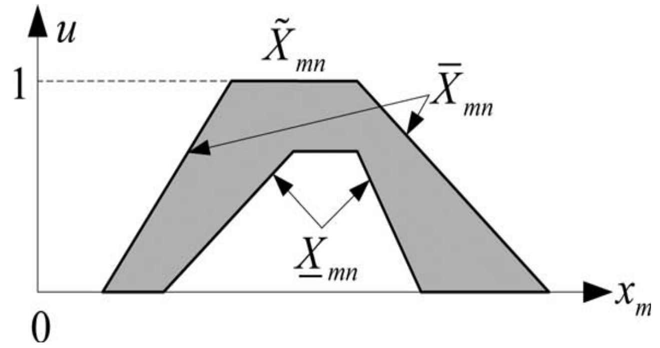


Fig. 1. Type-2 fuzzy set

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 Minputs { x_m} m=1,2,...,Mand one output y. Assume the mth input has N_mMFs in its universe of discourse $\mathbb{X}_m$. Denote the nth MF in the mth input domain as $\tilde{X}_{mn}$. A complete rulebase with all possible combinations of the input fuzzy system consists of $K = \prod_{m=1}^{M} N_m$ rules in the form of:

$$\begin{aligned}&\tilde{R}^k\text{: IF } x_1 \text{ is } \tilde{X}_{1,n_{1k}} \text{ and ...and } x_M \text{ is } \tilde{X}_{M,n_{Mk}}\\ &[\underline{y}_k, \overline{y}_k],\, n_{ik} = 1,2,\ldots,N_i,\, k = 1,2,\ldots,K\end{aligned} \tag{1}$$

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 kth 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 $\boldsymbol{x}$ = (x1, x2, . . . , xM) is computed as:

$$y(\mathrm{x}) = \frac{y_l(\mathrm{x}) + y_r(\mathrm{x})}{2} \tag{2}$$

Where

$$y_l(\mathrm{x}) = \min_{\forall f_k \in [\underline{f}_k, \overline{f}_k]} \frac{\sum_{k=1}^{K} f_k \underline{y}_k}{\sum_{k=1}^{K} f_k} = \frac{\sum_{k=1}^{k_l} \overline{f}_k \underline{y}_k + \sum_{k=k_l+1}^{K} \underline{f}_k \underline{y}_k}{\sum_{k=1}^{k_l} \overline{f}_k + \sum_{k=k_l+1}^{K} \underline{f}_k} \tag{3}$$

$$y_r(\mathrm{x}) = \max_{\forall f_k \in [\underline{f}_k, \overline{f}_k]} \frac{\sum_{k=1}^{K} f_k \underline{y}_k}{\sum_{k=1}^{K} f_k} = \frac{\sum_{k=1}^{k_r} \underline{f}_k \overline{y}_k + \sum_{k=k_r+1}^{K} \overline{f}_k \overline{y}_k}{\sum_{k=1}^{k_r} \underline{f}_k + \sum_{k=k_r+1}^{K} \overline{f}_k} \tag{4}$$

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 [38].

2.3 GAs

Genetic algorithms (GAs) are numerical optimization algorithms inspired by both natural selection and genetics. We can also say that the genetic algorithm is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the “fitness”[15]. The method is a general one, capable of being applied to an extremely wide range of problems. The algorithms are simple to understand and the required computer code easy to write. GAs were in essence proposed by John Holland in the 1960's. His reasons for developing such algorithms went far beyond the type of problem solving with which this work is concerned. His 1975 book, Adaptation in Natural and Artificial Systems is particularly worth reading for its visionary approach. More recently others, for example De Jong, in a paper entitled Genetic Algorithms are NOT Function Optimizers , have been keen to remind us that GAs are potentially far more than just a robust method for estimating a series of unknown parameters within a model of a physical system[5]. A typical algorithm might consist of the following:

1. Start with a randomly generated population of n l–bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness *f*(x) of each chromosome *x* in the population.
3. Repeat the following steps until n offspring have been created:

- Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done "with replacement," meaning that the same chromosome can be selected more than once to be-come a parent.

- With probability P_c (the "crossover probability" or "crossover rate"), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offspring that are exact copies of their respective parents. (Note that here the crossover rate is defined to be the probability that two parents will cross over in a single point. There are also "multipoint crossover" versions of the GA in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.)
- Mutate the two offspring at each locus with probability P_m (the mutation probability or mutation rate), and place the resulting chromosomes in the new population. If n is odd, one new population member can be discarded at random.
- Replace the current population with the new population.
- Go to step 2 [27].

Some of the advantages of a GA include:

- Optimizes with continuous or discrete variables,
- Doesn't require derivative information,
- Simultaneously searches from a wide sampling of the cost surface,
- Deals with a large number of variables,
- Is well suited for parallel computers,
- Optimizes variables with extremely complex cost surfaces (they can jump out of a local minimum),
- Provides a list of optimal values for the variables, not just a single solution,
- Codification of the variables so that the optimization is done with the encoded variables, and
- Works with numerically generated data, experimental data, or analytical functions [13].

3 Intelligent Control of Nonlinear Dynamic Plants Using a Hierarchical Modular Approach and Type-2 Fuzzy Logic

The main goal of this work is to develop type-2 fuzzy systems for automatic control of nonlinear dynamic plants using a fuzzy granular approach and bio-inspired optimization; our work scheme is shown in Fig.2.

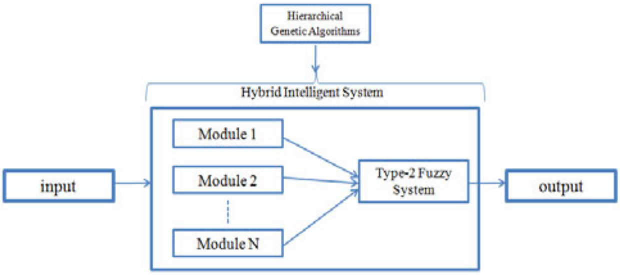


Fig. 2. Proposed modular approach for control

The use of Type-2 granular models is a contribution of this paper to improve the solution of the control problem that is going to be considered, since it divides the problem in modules for the different types of control and this model will receive the signal for further processing and perform adequate control. We can use this architecture in many cases to develop each controller separately. We can see in Fig.3 an example that how we can use this architecture in the area of control. In this example the fuzzy logic control has inputs 1 to n and outputs are also 1 to n. When we have more than one thing to control we can use type-1 fuzzy logic in each controller and then when we will have the outputs, we can then use the outputs and implement a type-2 fuzzy system to combine these outputs, and finally optimize the fuzzy system with the genetic algorithm.

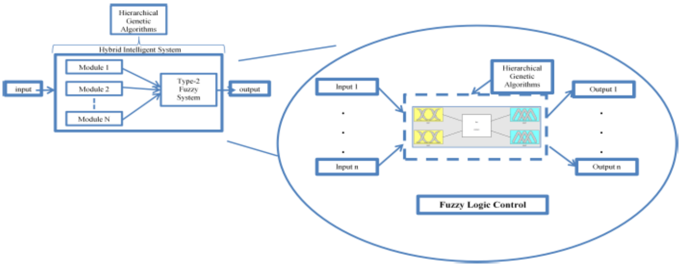


Fig. 3. Proposed granular fuzzy system

4 Automatic Design of Fuzzy Systems for Control of Aircraft Dynamic Systems with Genetic Optimization

We consider the problem of aircraft control as one case to illustrate the proposed approach. Over time the airplanes have evolved and at the same time there has been work on improving their techniques for controlling their flight and avoid accidents as much as possible. For this reason, we are considering in this paper the implementation of a system that controls the horizontal position of the aircraft. We created the fuzzy system to perform longitudinal control of the aircraft and then we used a simulation tool to test the fuzzy controller under noisy conditions. We designed the fuzzy controller with the purpose of maintaining the stability in horizontal flight of the aircraft by controlling only the movement of the elevators. We also use a genetic algorithm to optimize the fuzzy logic control design.

4.1 Problem Description

The purpose of this work was to develop an optimal fuzzy system for automatic control to maintain the aircraft in horizontal flight. The goal was to create the fuzzy system to perform longitudinal control of the aircraft and also to use a simulation tool to test the fuzzy controller with noise. The main goal was to achieve the stability in horizontal flight of the aircraft by controlling only the movement of the elevators.

4.2 PID and Fuzzy System for Longitudinal Control

If we want to use the longitudinal control we need to use 3 elements, which are: Stick: The lever of the pilot. Moving the control stick back-wards (toward the pilot) will rise the nose of the plane, and if push forward there is a lowering of the nose of the plane. Angle of attack (α).Rate of elevation (q): The speed at which an aircraft climbs. We need the above mentioned elements to perform elevator control. The comparison of the control system was carried out by first using the PID controller for longitudinal control and then we compared the results obtained with the same plant but using a fuzzy controller that was created and eventually carried out the simulation of the 2 controllers and the comparison of the results of fuzzy control with respect to PID control. The fuzzy system has 3 inputs (stick, angle of attack and rate of elevation) and 1 output (elevators). The fuzzy system that we used as a controller has 3 membership functions for each of the inputs and 3 membership functions of the output. We worked with different types of membership functions, such as the Gaussian, Bell, Trapezoidal and Triangular.

4.3 Simulation Results

In this section we present the results obtained when performing the tests using the simulation plant with the PID and Fuzzy controllers. It also presents the results obtained by optimizing the fuzzy system with a genetic algorithm. The first simulation was performed with the PID controller and we obtained the elevators behavior. We obtained an average elevator angle of 0.2967. Once the simulation results with the PID Controller were obtained, we proceeded with our Fuzzy Controller using the fuzzy system that was created previously. The simulations were carried out with different types of membership functions and the results that were obtained are shown in Table 1.

Table 1. Results for simulation plant with a type-1 fuzzy controller

Membership functions	Trapezoidal	Triangular	Gauss	Bell
Errors comparing with PID	0 .1094	0.1131	0.1425	0.1222
Comments	Fast Simulation	Less Fast Simulation	Slow simulation in comparison with previous	Slow simulation in comparison with previous

Having obtained the above results, we used a genetic algorithm to optimize the membership functions of the fuzzy system and after implementing the genetic algorithm we obtained the optimized results shown in Table 2.

Table 2. Results for the simulation plant with the fuzzy controller optimized by a Genetic Algorithm

Genetic Algorithm	**Error with respect to PID**
Using **Trapezoidal** membership functions	0.0531
Using **Gauss** membership functions	0.084236395
Using **Bell** membership functions	0.0554
Using **Triangular** membership functions	0.0261

Given the above results we can see that better results were obtained using genetic algorithms and in particular the best result was using Membership functions of triangular type. When we used the genetic algorithm the best result that we obtained was when we worked using triangular membership functions because we obtained an error of 0.0261.When we apply the Genetic Algorithm using a sine wave as a reference in our simulation plant (see Table 3) we could observe differences between the simulations. As we mentioned before we used 4 types of membership functions, such as bell, Gauss, trapezoidal and triangular. At the time of carrying out the simulation, the error was 0.004 using bell membership functions, as we can appreciate this is the better result. The decrease of error is because when we work with sine wave at the time of carrying out the simulation, our plant does not have many problems for this type of waveform and that is because the sine wave is easier to follow (higher degree of continuity).When we work using square wave we have more complex behavior because this kind of wave is more difficult. To consider a more challenging problem we decided to continue working with square wave and in this form improve our controller. We were also interested in improving the controller by adding noise to the plant. We decided to use Gaussian noise to simulate uncertainty in the control process. The Gaussian Noise Generator block generates discrete-time white Gaussian noise. Results with more noise are shown in Table 4.

Table 3. Results for simulation plant with fuzzy controller and Genetic Algorithm

Genetic Algorithm	**Error with respect to PID**
Using **Trapezoidal** membership functions	0.0491
Using **Gauss** membership functions	0.0237
Using **Triangular** membership functions	0.0426
Using **Bell** membership functions	0.004

Table 4. Results for the simulation plant with a fuzzy controller and Gaussian noise (Type-2 and Type-1)

Membership functions	**Noise Level**					
	84	123	580	1200	2500	5000
Triangular	0.1218	0.1191	0.1228	0.1201	0.1261	0.1511
Trapezoidal	0.1182	0.1171	0.1156	0.1196	0.1268	0.1415
Gauss	0.1374	0.1332	0.1356	0.1373	0.1365	0.1563
Bell	0.119	0.1172	0.1171	0.1203	0.1195	0.1498
Type-2 Triangular	0.1623	0.1614	0.1716	0.163	0.1561	0.1115

In this case a type-2 fuzzy system (last row) produces a better result when the noise level is high. In Table 4 we can observe that in many cases the type-1 provided better results than type-2. But when we raise the noise level the type-2 fuzzy system obtained better results as it supports higher levels of noise.

5 Hierarchical Genetic Algorithm for Optimal Type-2 Fuzzy System Design in the Shower Control

In this case we propose an algorithm to optimize a fuzzy system to control the Temperature in the Shower benchmark problem, in this application the fuzzy controller has two inputs: the water temperature and the flow rate. The controller uses these inputs to set the position of the hot and cold valves. In this part the genetic algorithm optimized the fuzzy system for control.

5.1 Problem Description

The problem was of developing a genetic algorithm to optimize the parameters of a fuzzy system that can be applied in the fuzzy logic areas. The main goal was to achieve the best result in each application, in our case fuzzy control of the shower. We started to work with different membership functions in these cases and after performing the tests finally we took the best result. The genetic algorithm can change the number of inputs and outputs depending on that we need it. The Chromosome for this case is shown in fig.4.

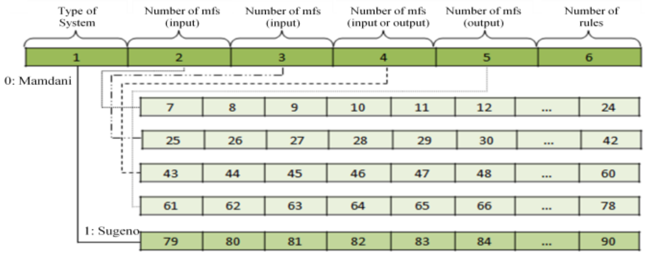


Fig. 4. Chromosome of the Genetic Algorithm

5.2 Fuzzy Control

In this case we realized the simulation with the Simulink plant in the Matlab programming language. The problem was to improve temperature control in a shower example the original fuzzy system has two inputs to the fuzzy controller: the water temperature and the flow rate. The controller uses these inputs to set the position of the hot and cold valves. When we simulated the type-2 fuzzy system the best result that we obtained was 0.000096, and in the same problem but using type-1 we obtained 0.05055. This shows that type-2 fuzzy control can outperform type-1 for this problem. The best fuzzy system that we obtained in fuzzy control is shown in Figure 5.

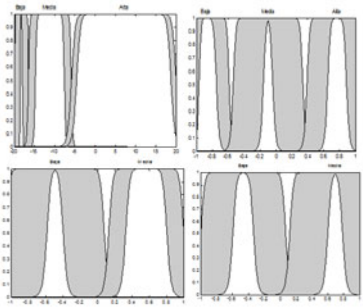


Fig. 5. Type-2 Fuzzy system for control

6 Conclusions

We use two benchmark problems and based on the obtained results we can say that to achieve control of the present problems, type-2 fuzzy logic is a good alternative to achieve good control. When we worked with a type-1 fuzzy system we obtained good results, but if we want to work with noise the previous good results will not be so good, in this case we need to work with type-2 and with this we obtained better results and also using a genetic algorithm to optimize the fuzzy system. When we have a problem for example to control the flight of an airplane we need to control 3 different controllers. In this case the fuzzy granular method is of great importance, because we want to control the flight of an airplane completely. We want to use a type-1 fuzzy system in each controller and then use a type-2 fuzzy system to combine the outputs of the type-1 fuzzy systems and implement the concept of granularity and with this method we hope to obtain a better result in this problem.

References

1. Abusleme, H., Angel, C.: Fuzzy control of an unmanned flying vehicle, Ph.d. Thesis, Ponti-ficia University of Chile (2000)
2. Bargiela, A., Wu, P.: Granular Computing: An Introduction. Kluwer Academic Publish, Dordercht (2003)
3. Bargiela, A., Wu, P.: The roots of Granular Computing. In: GrC 2006, pp. 806–809. IEEE (2006)
4. Blakelock, J.: Automatic Control of Aircraft and Missiles. Prentice-Hall (1965)
5. Coley, A.: An Introduction to Genetic Algorithms for Scientists and Engineers. World Scientific (1999)
6. The 2011 IEEE Internaional Confenrece on Granular Computing Sapporo, GrC 2011, Japan, August 11-13. IEEE Computer Society (2011)
7. Dorf, R.: Modern Control Systems. Addison-Wesley Pub. Co. (1997)
8. Dwinnell, J.: Principles of Aerodynamics. McGraw-Hill Company (1929)
9. Engelen, H., Babuska, R.: Fuzzy logic based full-envelope autonomous flight control for an atmospheric re-entry spacecraft. Control Engineering Practice Journal 11(1), 11–25 (2003)
10. Federal Aviation Administration, Airplane Flying Handbook, U.S. Department of Transportation Federal Aviation Administration (2007)
11. Federal Aviation Administration. Pilot's Handbook of Aeronautical Knowledge, U.S. Department of Transportation Federal Aviation Administration (2008)
12. Gardner A.: U.S Warplanes The F-14 Tomcat, The Rosen Publishing Group (2003)
13. Gibbens, P., Boyle, D.: Introductory Flight Mechanics and Performance. University of Sydney, Australia. Paper (1999)
14. Goedel, K.: The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis with the Axioms of Set Theory. Princeton University Press, Princeton (1940)
15. Haupt R., Haupt S.: Practical Genetic Algorithm. Wiley-Interscience (2004)
16. Holmes, T.: US Navy F-14 Tomcat Units of Operation Iraqi Freedom, Osprey Publishing Limited (2005)
17. Jamshidi, M., Vadiee, N., Ross, T.: Fuzzy Logic and Control: Software and Hardware Applications, vol. 2. Prentice-Hall, University of New Mexico (1993)

18. Kadmiry, B., Driankov, D.: A fuzzy flight controller combining linguistic and model based fuzzy control. Fuzzy Sets and Systems Journal 146(3), 313–347 (2004)
19. Karnik, N., Mendel, J.: Centroid of a type-2 fuzzy set. Information Sciences 132, 195–220 (2001)
20. Keviczky, T., Balas, G.: Receding horizon control of an F-16 aircraft: A comparative study. Control Engineering Practice Journal 14(9), 1023–1033 (2006)
21. Liu, M., Naadimuthu, G., Lee, E.S.: Trayectory tracking in aircraft landing operations management using the adaptive neural fuzzy inference system. Computers & Mathematics with Applications Journal 56(5), 1322–1327 (2008)
22. McLean D.: Automatic Flight Control System. Prentice Hall (1990)
23. McRuer, D., Ashkenas, I., Graham, D.: Aircraft Dynamics and Automatic Control. Princeton University Press (1973)
24. Melin, P., Castillo, O.: Intelligent control of aircraft dynamic systems with a new hybrid neuro- fuzzy–fractal Approach. Journal Information Sciences 142(1) (May 2002)
25. Melin, P., Castillo, O.: Adaptive intelligent control of aircraft systems with a hybrid approach combining neural networks, fuzzy logic and fractal theory. Journal of Applied Soft computing 3(4) (December 2003)
26. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
27. Mitchell, M.: An Introduction to Genetic Algorithms. Massachusetts Institute of Technology (1999)
28. Morelli, E.A.: Global Nonlinear Parametric Modeling with Application to F-16 Aerodynamics, NASA Langley Research Center, Hampton, Virginia (1995)
29. Nelson, R.: Flight Stability and automatic control, 2nd edn. Department of Aerospace and Mechanical Engineering, University of Notre Dame., McGraw Hill (1998)
30. Pedrycz, W., Skowron, A., et al.: Handbook granular computing. Wiley Interscience, New York (2008)
31. Rachman, E., Jaam, J., Hasnah, A.: Non-linear simulation of controller for longitudinal control augmentation system of F-16 using numerical approach. Information Sciences Journal 164(1-4), 47–60 (2004)
32. Reiner, J., Balas, G., Garrard, W.: Flight control design using robust dynamic inversion and time- scale separation. Automatic Journal 32(11), 1493–1504 (1996)
33. Sanchez, E., Becerra, H., Velez, C.: Combining fuzzy, PID and regulation control for an autonomous mini-helicopter. Journal of Information Sciences 177(10), 1999–2022 (2007)
34. Sefer, K., Omer, C., Okyay, K.: Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. Expert Systems with Applications Journal 37(2), 1229–1234 (2010)
35. Song, Y., Wang, H.: Design of Flight Control System for a Small Unmanned Tilt Rotor Air-craft. Chinese Journal of Aeronautics 22(3), 250–256 (2009)
36. Walker, D.J.: Multivariable control of the longitudinal and lateral dynamics of a fly by wire helicopter. Control Engineering Practice 11(7), 781–795 (2003)
37. Wu, D.: A Brief Tutorial on Interval Type-2 Fuzzy Sets and Systems (July 22, 2010)
38. Wu, D., Jerry, M.: On the Continuity of Type-1 and Interval Type-2 Fuzzy Logic Systems. IEEE T. Fuzzy Systems 19(1), 179–192 (2011)
39. 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)
40. 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)

No-Free-Lunch Result for Interval and Fuzzy Computing: When Bounds Are Unusually Good, Their Computation Is Unusually Slow

Martine Ceberio and Vladik Kreinovich

University of Texas at El Paso, Computer Science Dept., El Paso, TX 79968, USA
{mceberio,vladik}@utep.edu

Abstract. On several examples from interval and fuzzy computations and from related areas, we show that when the results of data processing are unusually good, their computation is unusually complex. This makes us think that there should be an analog of Heisenberg's uncertainty principle well known in quantum mechanics: when we an unusually beneficial situation in terms of results, it is not as perfect in terms of computations leading to these results. In short, nothing is perfect.

1 First Case Study: Interval Computations

Need for data processing. In science and engineering, we want to *understand* how the world works, we want to *predict* the results of the world processes, and we want to *design* a way to control and change these processes so that the results will be most beneficial for the humankind.

For example, in meteorology, we want to know the weather now, we want to predict the future weather, and – if, e.g., floods are expected, we want to develop strategies that would help us minimize the flood damage.

Usually, we know the equations that describe how these systems change in time. Based on these equations, engineers and scientists have developed algorithms that enable them to predict the values of the desired quantities – and find the best values of the control parameters. As input, these algorithms take the current and past values of the corresponding quantities.

For example, if we want to predict the trajectory of the spaceship, we need to find its current location and velocity, the current position of the Earth and of the celestial bodies, then we can use Newton's equations to find the future locations of the spaceship.

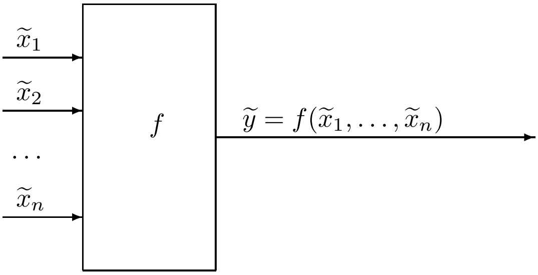
In many situations – e.g., in weather prediction – the corresponding computations require a large amount of input data and a large amount of computations steps. Such computations (*data processing*) are the main reason why computers were invented in the first place – to be able to perform these computations in reasonable time.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 13–23, 2011.

Need to take input uncertainty into account. In all the data processing tasks, we start with the current and past values $x_1, \ldots, x_n$ of some quantities, and we use a known algorithm $f(x_1, \ldots, x_n)$ to produce the desired result $y = f(x_1, \ldots, x_n)$.

The values x_i come from measurements, and measurements are never absolutely accurate: the value $\widetilde{x}_i$ that we obtained from measurement is, in general, different from the actual (unknown) value x_i of the corresponding quantity. For example, if the clock shows 12:20, it does not mean that the time is *exactly* 12 hours, 20 minutes and 00.0000 seconds: it may be a little earlier or a little later than that.

As a result, in practice, we apply the algorithm f not to the actual values x_i, but to the *approximate* values $\widetilde{x}_i$ that come from measurements:


$\widetilde{x}_1$
$\widetilde{x}_2$
...
$\widetilde{x}_n$
f
$\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$

So, instead of the ideal value $y = f(x_1, \ldots, x_n)$, we get an approximate value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$. A natural question is: how do approximation errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ affect the resulting error $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$? Or, in plain words, how to take input uncertainty into account in data processing?

From probabilistic to interval uncertainty. [18] Manufacturers of the measuring instruments provide us with bounds Δ_i on the (absolute value of the) measurement errors: $|\Delta x_i| \leq \Delta_i$. If now such upper bound is known, then the device is *not* a measuring instrument.

For example, a street thermometer may show temperature that is slightly different from the actual one. Usually, it is OK if the actual temperature is +24 but the thermometer shows +22 – as long as the difference does not exceed some reasonable value Δ. But if the actual temperature is +24 but the thermometer shows −5, any reasonable person would return it to the store and request a replacement.

Once we know the measurement result $\widetilde{x}_i$, and we know the upper bound Δ_i on the measurement error, we can conclude that the actual (unknown) value x_i belongs to the interval $[\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. For example, if the measured temperature is $\widetilde{x}_i = 22$, and the manufacturer guarantees the accuracy $\Delta_i = 3$, this means that the actual temperature is somewhere between $\widetilde{x}_i - \Delta_i = 22 - 3 = 19$ and $\widetilde{x}_i + \Delta_i = 22 + 3 = 25$.

Often, in addition to these bounds, we also know the *probabilities* of different possible values Δx_i within the corresponding interval $[-\Delta_i, \Delta_i]$. This is how uncertainty is usually handled in engineering and science – we assume that we know the probability distributions for the measurement errors Δx_i (in most cases, we assume that this distribution is normal), and we use this information to describe the probabilities of different values of Δy. However, there are two important situations when we do not know these probabilities:

- cutting-edge measurements, and
- cutting-cost manufacturing.

Indeed, how do we determine the probabilities? Usually, to find the probabilities of different values of the measurement error $\Delta x_i = \widetilde{x}_i - x_i$, we bring our measuring instrument to a lab that has a "standard" (much more accurate) instrument, and compare the results of measuring the same quantity with two different instruments: ours and a standard one. Since the standard instrument is much more accurate, we can ignore its measurement error and assume that the value X_i that it measures is the actual value: $X_i \approx x_i$. Thus, the difference $\widetilde{x}_i - X_i$ between the two measurement results is practically equal to the measurement error $\Delta x_i = \widetilde{x}_i - x_i$. So, when we repeat this process several times, we get a histogram from which we can find the probability distribution of the measurement errors.

However, in the above two situations, this is not done. In the case of cutting-edge measurements, this is easy to explain. For example, if we want to estimate the measurement errors of the measurement performed by a Hubble space telescope (or by the newly built CERN particle collider), it would be nice to have a "standard", five times more accurate telescope floating nearby – but Hubble is the best we have. In manufacturing, in principle, we can bring every single sensor to the National Institute of Standards and determine its probability distribution – but this would cost a lot of money: most sensors are very cheap, and their "calibration" using the expensive super-precise "standard" measuring instruments would cost several orders of magnitude more. So, unless there is a strong need for such calibration – e.g., if we manufacture a spaceship – it is sufficient to just use the upper bound on the measurement error.

In both situations, after the measurements, the only information that we have about the actual value of x_i is that this value belongs to the interval $[\underline{x}_i, \overline{x}_i] = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

Different possible values x_i from the corresponding intervals lead, in general, to different values of $y = f(x_1, \ldots, x_n)$. It is therefore desirable to find the range of all possible values of y, i.e., the set

$$\mathbf{y} = [\underline{y}, \overline{y}] = \{f(x_1, \ldots, x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]\}.$$

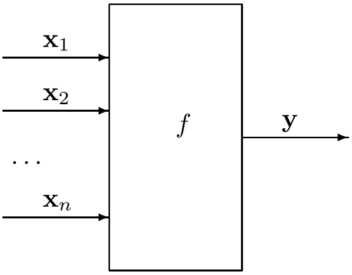
(Since the function $f(x_1, \ldots, x_n)$ is usually continuous, its range is the interval.) Thus, we arrive at the same *interval computations* problem; see, e.g., [6,7,15].

The main problem. We are given:

- an integer n;
- n intervals $\mathbf{x}_1 = [\underline{x}_1, \overline{x}_1]$, ..., $\mathbf{x}_n = [\underline{x}_n, \overline{x}_n]$, and
- an algorithm $f(x_1, \ldots, x_n)$ which transforms n real numbers into a real number $y = f(x_1, \ldots, x_n)$.

We need to compute the endpoints $\underline{y}$ and $\overline{y}$ of the interval

$$\mathbf{y} = [\underline{y}, \overline{y}] = \{f(x_1, \ldots, x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n]\}.$$



In general, the interval computations problem is NP-hard. It is known that in general, the problem of computing the exact range $\mathbf{y}$ is NP-hard; see, e.g., [13]. Moreover, it is NP-hard even if we restrict ourselves to quadratic functions $f(x_1, \ldots, x_n)$ – even to the case when we only consider a very simple quadratic function: a sample variance [2,3]:

$$f(x_1, \ldots, x_n) = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n} \cdot \sum_{i=1}^{n} x_i \right)^2 .$$

NP-hard means, crudely speaking, that it is not possible to have an algorithm that would always compute the exact range in reasonable time.

Case of small measurement errors. In many practical situations, the measurement errors are relatively small, i.e., we can safely ignore terms which are quadratic or higher order in terms of these errors. For example, if the measurement error is 10%, its square is 1% which is much smaller than 10%. In such situations, it is possible to have an efficient algorithm for computing the desired range.

Indeed, in such situations, we can simplify the expression for the desired error

$$\Delta y = \widetilde{y} - y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(x_1, \ldots, x_n) =$$

$$f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n)$$

if we expand the function f in Taylor series around the point $(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ and restrict ourselves only to linear terms in this expansion. As a result, we get the expression

$$\Delta y = c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n,$$

where by c_i, we denoted the value of the partial derivative $\partial f/\partial x_i$ at the point $(\widetilde{x}_1, \ldots, \widetilde{x}_n)$:

$$c_i = \frac{\partial f}{\partial x_i}_{|(\widetilde{x}_1, \ldots, \widetilde{x}_n)}.$$

In the case of interval uncertainty, we do not know the probability of different errors Δx_i; instead, we only know that $|\Delta x_i| \le \Delta_i$. In this case, the above sum attains its largest possible value if each term $c_i \cdot \Delta x_i$ in this sum attains the largest possible value:

- If $c_i \ge 0$, then this term is a monotonically non-decreasing function of Δx_i, so it attains its largest value at the largest possible value $\Delta x_i = \Delta_i$; the corresponding largest value of this term is $c_i \cdot \Delta_i$.
- If $c_i < 0$, then this term is a decreasing function of Δx_i, so it attains its largest value at the smallest possible value $\Delta x_i = -\Delta_i$; the corresponding largest value of this term is $-c_i \cdot \Delta_i = |c_i| \cdot \Delta_i$.

In both cases, the largest possible value of this term is $|c_i| \cdot \Delta_i$, so, the largest possible value of the sum Δy is

$$\Delta = |c_1| \cdot \Delta_1 + \ldots + |c_n| \cdot \Delta_n.$$

Similarly, the smallest possible value of Δy is $-\Delta$.

Hence, the interval of possible values of Δy is $[-\Delta, \Delta]$, with Δ defined by the above formula.

How do we compute the derivatives? If the function f is given by its analytical expression, then we can simply explicitly differentiate it, and get an explicit expression for its derivatives. This is the case which is typically analyzed in textbooks on measurement theory; see, e.g., [18].

In many practical cases, we do not have an explicit analytical expression, we only have an *algorithm* for computing the function $f(x_1, \ldots, x_n)$, an algorithm which is too complicated to be expressed as an analytical expression.

When this algorithm is presented in one of the standard programming languages such as Fortran or C, we can apply one of the existing analytical differentiation tools (see, e.g., [5]), and automatically produce a program which computes the partial derivatives c_i. These tools analyze the code and produce the differentiation code as they go.

In many other real-life applications, an algorithm for computing $f(x_1, \ldots, x_n)$ may be written in a language for which an automatic differentiation tool is not available, or a program is only available as an executable file, with no source code at hand. In such situations, when we have no easy way to analyze the code, the only thing we can do is to take this program as a *black box*: i.e., to apply it to different inputs and use the results of this application to compute the desired value Δ. Such black-box methods are based on the fact that, by definition, the derivative is a limit:

$$c_i = \frac{\partial f}{\partial x_i} =$$

$$\lim_{h\to 0} \frac{f(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i+h,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n) - f(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n)}{h}.$$

By definition, a limit means that when h is small, the right-hand side expression is close to the derivative – and the smaller h, the closer this expression to the desired derivative. Thus, to find the derivative, we can use this expression for some small h:

$$c_i \approx \frac{f(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i+h,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n) - f(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n)}{h}.$$

To find all n partial derivatives c_i, we need to call the algorithm for computing the function $f(x_1,\ldots,x_n)$ $n+1$ times:

- one time to compute the original value $f(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n)$ and
- n times to compute the perturbed values $f(\widetilde{x}_1,\ldots,\widetilde{x}_{i-1},\widetilde{x}_i+h,\widetilde{x}_{i+1},\ldots,\widetilde{x}_n)$ for $i=1,2,\ldots,n$.

So:

- if the algorithm for computing the function $f(x_1,\ldots,x_n)$ is feasible, finishes its computations in polynomial time T_f, i.e., in time which is bounded by a polynomial of the size n of the input,
- then the overall time needed to compute all n derivatives c_i is bounded by $(n+1)\cdot T_f$ and is, thus, also polynomial – i.e., feasible.

Cases when the resulting error is unusually small. In general, the resulting approximation error Δ is a linear function of the error bounds Δ_1, $\ldots$, Δ_n on individual (direct) measurements. In other words, the resulting approximation error is of the same order as the original bounds Δ_i. In this general case, the above technique (or appropriate faster techniques; see, e.g., [9,19]) provide a good estimate for Δ, an estimate with an absolute accuracy of order Δ_i^2 and thus, with a relative accuracy of order Δ_i.

There are usually good cases, when all (or almost all) linear terms in the linear expansion disappear: when the derivatives $c_i = \dfrac{\partial f}{\partial x_i}$ are equal to 0 (or close to 0) at the point $(\widetilde{x}_1,\ldots,\widetilde{x}_n)$. In this case, to estimate Δ, we must consider next terms in Taylor expansion, i.e., terms which are quadratic in Δ_i:

$$\Delta y = \widetilde{y} - y = f(\widetilde{x}_1,\ldots,\widetilde{x}_n) - f(x_1,\ldots,x_n) =$$

$$f(\widetilde{x}_1,\ldots,\widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1,\ldots,\widetilde{x}_n - \Delta x_n) =$$

$$f(\widetilde{x}_1,\ldots,\widetilde{x}_n) - \left(f(\widetilde{x}_1,\ldots,\widetilde{x}_n) + \frac{1}{2}\cdot\sum_{i=1}^n\sum_{j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}\cdot \Delta x_i \cdot \Delta x_j + \ldots \right) =$$

$$-\frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j} \cdot \Delta x_i \cdot \Delta x_j + \ldots$$

As a result, in such situations, the resulting approximation error is unusually small – it is proportional to Δ_i^2 instead of Δ_i. For example, when the measurement accuracy is $\Delta_i \approx 10\%$, usually, we have Δ of the same order 10%, but in this unusually good case, the approximation accuracy is of order $\Delta_i^2 \approx 1\%$ – an order of magnitude better.

When bounds are unusually good, their computation is unusually slow. In the above case, estimating Δ means solving an interval computations problem (of computing the range of a given function on given intervals) for a quadratic function $f(x_1, \ldots, x_n)$. We have already mentioned that, in contrast to the linear case when we have an efficient algorithm, the interval computation problem for quadratic functions is NP-hard. Thus, when bounds are unusually small, their computation is an unusually difficult task.

Discussion. The above observation us think that there should be an analog of Heisenberg's uncertainty principle (well known in quantum mechanics):

- when we an unusually beneficial situation in terms of results,
- it is not as perfect in terms of computations leading to these results.

In short, nothing is perfect.

Comment. Other examples – given below – seem to confirm this conclusion.

2 Second Case Study: Fuzzy Computations

Need for fuzzy computations. In some cases, in addition to (and/or instead of) measurement results x_i, we have expert estimates for the corresponding quantities. These estimates are usually formulated by using words from natural language, like "about 10". A natural way to describe such expert estimates is to use fuzzy techniques (see, e.g., [8,17]), i.e., to describe each such estimate as a *fuzzy number* X_i – i.e., as a function $\mu_i(x_i)$ that assigns, to each possible value x_i, a degree to which the expert is confident that this value is possible. This function is called a *membership function.*

Fuzzy data processing. When each input x_i is described by a fuzzy number X_i, i.e., by a membership function $\mu_i(x_i)$ that assigns, to every real number x_i, a degree to which this number is possible as a value of the i-th input, we want to find the fuzzy number Y that describes $f(x_1, \ldots, x_n)$. A natural way to define the corresponding membership function $\mu(y)$ leads to Zadeh's extension principle:

$$\mu(y) = \sup\{\min(\mu_1(x_1), \ldots, \mu_n(x_n)) : f(x_1, \ldots, x_n) = y\}.$$

Fuzzy data processing can be reduced to interval computations. It is known that from the computational viewpoint, the application of this formula can be reduced to interval computations.

Specifically, for each fuzzy set with a membership function $\mu(x)$ and for each $\alpha \in (0,1]$, we can define this set's *α-cut* as $\mathcal{X}(\alpha) \stackrel{\text{def}}{=} \{x : \mu(x) \geq \alpha\}$. Vice versa, if we know the α-cuts for all α, we, for each x, can reconstruct the value $\mu(x)$ as the largest value α for which $x \in \mathcal{X}(\alpha)$. Thus, to describe a fuzzy number, it is sufficient to find all its α-cuts.

It is known that when the inputs $\mu_i(x_i)$ are fuzzy numbers, and the function $y = f(x_1, \ldots, x_n)$ is continuous, then for each α, the α-cut $\mathcal{Y}(\alpha)$ of y is equal to the range of possible values of $f(x_1, \ldots, x_n)$ when $x_i \in \mathcal{X}_i(\alpha)$ for all i:

$$\mathcal{Y}(\alpha) = f(\mathcal{X}_1(\alpha), \ldots, \mathcal{X}_n(\alpha)) = \{f(x_1, \ldots, x_n) : x_1 \in \mathcal{X}_1(\alpha), \ldots, x_n \in \mathcal{X}_n\};$$

see, e.g., [1,8,16,17]. So, if we know how to solve our problem under interval uncertainty, we can also solve it under fuzzy uncertainty – e.g., by repeating the above interval computations for $\alpha = 0, 0.1, \ldots, 0.9, 1.0$.

When bounds are unusually good, their computation is unusually slow. Because of the above reduction, the conclusion about interval computations can be extended to fuzzy computations:

- when the resulting bounds are unusually good,
- their computation is unusually difficult.

3 Third Case Study: When Computing Variance under Interval Uncertainty Is NP-Hard

Computing the range of variance under interval uncertainty is NP-hard: reminder. The above two examples are based on the result that computing the range of a quadratic function under interval uncertainty is NP-hard. Actually, as we have mentioned, even computing the range $[\underline{V}, \overline{V}]$ of the variance $V(x_1, \ldots, x_n)$ on given intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is NP-hard [2,3]. Specifically, it turns out that while the lower endpoint $\underline{V}$ can be computed in polynomial time, computing the upper endpoint $\overline{V}$ is NP-hard.

Let us move analysis deeper. Let us check when we should expect the most beneficial situation – with small $\overline{V}$ – and let us show that in this case, computing $\overline{V}$ is the most difficult task.

When we can expect the variance to be small. By definition, the variance $V = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} (x_i - E)^2$ describes the average deviation of its values from the mean $E = \dfrac{1}{n} \cdot \sum\limits_{i=1}^{n} x_i$. The smallest value of the variance V is attained when all the values from the sample are equal to the mean E, i.e., when all the values in the sample are equal $x_1 = \ldots = x_n$.

In the case of interval uncertainty, it is thus natural to expect that the variance is small if it is possible that all values x_i are equal, i.e., if all n intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ have a common point.

In situations when we expect small variance, its computation is unusually slow. Interestingly, NP-hardness is proven, in [2,3], exactly on the example of n intervals that all have a common intersection – i.e., on the example when we should expect the small variance.

Moreover, if the input intervals do not have a common non-empty intersection – e.g., if there is a value C for which every collection of C intervals have an empty intersection – then it is possible to have a feasible algorithm for computing the range of the variance [2,3,4,10,11,12].

Discussion. Thus, we arrive at the same conclusion as in the above cases:

- when we an unusually beneficial situation in terms of results,
- it is not as perfect in terms of computations leading to these results.

4 Fourth Case Study: Kolmogorov Complexity

Need for Kolmogorov complexity. In many application areas, we need to compress data (e.g., an image). The original data can be, in general, described as a string x of symbols. What does it mean to compress a sequence? It means that instead of storing the original sequence, we store a compressed data string and a program describing how to un-compress the data. The pair consisting of the data and the un-compression program can be viewed as a single program p which, when run, generates the original string x. Thus, the quality of a compression can be described as the length of the shortest program p that generates x. This shortest length is known as *Kolmogorov complexity* $K(x)$ of the string x; see, e.g., [14]:

$$K(x) \stackrel{\text{def}}{=} \min\{\text{len}(p) : p \text{ generates } x\}.$$

In unusually good situations, computations are unusually complex. The smaller the Kolmogorov complexity $K(x)$, the more we can compress the original sequence x. It turns out (see, e.g., [14]) that, for most strings, the Kolmogorov complexity $K(x)$ is approximately equal to their length – and can, thus, be efficiently computed (as long as we are interested in the approximate value of $K(x)$, of course). These strings are what physicists would call *random.*

However, there are strings which are not random, strings which can be drastically compressed. It turns out that computing $K(x)$ for such strings is difficult: there is no algorithm that would, given such a string x, compute its Kolmogorov complexity (even approximately) [14]. This result confirms our general conclusion that:

- when situations are unusually good,
- computations are unusually complex.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 and DUE-0926721 and by Grant 1 T36 GM078000-01 from the National Institutes of Health.

The authors are thankful to Didier Dubois for valuable discussions, and to the anonymous referees for valuable suggestions.

References

1. Dubois, D., Prade, H.: Operations on fuzzy numbers. International Journal of Systems Science 9, 613–626 (1978)
2. Ferson, S., Ginzburg, L., Kreinovich, V., Longpré, L., Aviles, M.: Computing variance for interval data is NP-hard. ACM SIGACT News 33(2), 108–118 (2002)
3. Ferson, S., Ginzburg, L., Kreinovich, V., Longpré, L., Aviles, M.: Exact bounds on finite populations of interval data. Reliable Computing 11(3), 207–233 (2005)
4. Ferson, S., Kreinovich, V., Hajagos, J., Oberkampf, W., Ginzburg, L.: Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty, Sandia National Laboratories, Report SAND2007-0939 (May 2007)
5. Griewank, A., Walter, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. SIAM Publ., Philadelphia (2008)
6. Interval computations website, http://www.cs.utep.edu/interval-comp
7. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis, with Examples in Parameter and State Estimation. In: Robust Control and Robotics. Springer, London (2001)
8. Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice Hall, Upper Saddle River (1995)
9. Kreinovich, V., Ferson, S.: A new Cauchy-based black-box technique for uncertainty in risk analysis. Reliability Engineering and Systems Safety 85(1–3), 267–279 (2004)
10. Kreinovich, V., Longpré, L., Starks, S.A., Xiang, G., Beck, J., Kandathi, R., Nayak, A., Ferson, S., Hajagos, J.: Interval versions of statistical techniques, with applications to environmental analysis, bioinformatics, and privacy in statistical databases. Journal of Computational and Applied Mathematics 199(2), 418–423 (2007)
11. Kreinovich, V., Xiang, G., Starks, S.A., Longpré, L., Ceberio, M., Araiza, R., Beck, J., Kandathi, R., Nayak, A., Torres, R., Hajagos, J.: Towards combining probabilistic and interval uncertainty in engineering calculations: algorithms for computing statistics under interval uncertainty, and their computational complexity. Reliable Computing 12(6), 471–501 (2006)
12. Kreinovich, V., Xiang, G.: Fast algorithms for computing statistics under interval uncertainty: an overview. In: Huynh, V.-N., Nakamori, Y., Ono, H., Lawry, J., Kreinovich, V., Nguyen, H.T. (eds.) Interval/Probabilistic Uncertainty and Non-Classical Logics, pp. 19–31. Springer, Heidelberg (2008)
13. Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P.: Computational Complexity and Feasibility of Data Processing and Interval Computations. Kluwer, Dordrecht (1997)
14. Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer, Heidelberg (2008)
15. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM Press, Philadelphia (2009)

16. Nguyen, H.T., Kreinovich, V.: Nested intervals and sets: concepts, relations to fuzzy sets, and applications. In: Kearfott, R.B., Kreinovich, V. (eds.) Applications of Interval Computations, pp. 245–290. Kluwer, Dordrecht (1996)
17. Nguyen, H.T., Walker, E.A.: A First Course in Fuzzy Logic. Chapman & Hall/CRC, Boca Raton (2006)
18. Rabinovich, S.: Measurement Errors and Uncertainties: Theory and Practice. Springer, New York (2005)
19. Trejo, R., Kreinovich, V.: Error estimations for indirect measurements: randomized vs. deterministic algorithms for 'black-box' programs. In: Rajasekaran, S., Pardalos, P., Reif, J., Rolim, J. (eds.) Handbook on Randomized Computing, pp. 673–729. Kluwer (2001)

Intelligent Robust Control of Dynamic Systems with Partial Unstable Generalized Coordinates Based on Quantum Fuzzy Inference

Andrey Mishin[1] and Sergey Ulyanov[2]

[1] Dubna International University of Nature,
Society, and Man «Dubna”
[2] PronetLabs, Moscow
andrmish@yandex.ru, ulyanovsv@mail.ru

Abstract. This article describes a new method of quality control dynamically unstable object based on quantum computing. This method enables to control object in unpredicted situations with incomplete information about the structure of the control object. The efficiency over other methods of intelligent control is shown on the benchmark with partial unstable generalized coordinates as stroboscopic robotic manipulator.

Keywords: quantum fuzzy inference, control in unpredicted situations, robustness, intelligent control, quantum algorithms.

1 Introduction

The possibility of control unstable technical objects has considered for a long time. But practical importance controlling such objects has appeared relatively recent. The fact is unstable control objects (CO) have a lot of useful qualities (e.g. high-speed performance); it is possible if this objects properly controlled. But in case of failure of control of unstable object can represent a significant threat. In this kind of situations can apply the technology of computational intelligence, such as soft computing (including neural networks, genetic algorithms, fuzzy logic, and etc.) The advantage of intelligent control system is possibility to achieve the control goal in the presence of incomplete information about CO functional. The basis of any intelligent control system (ICS) is knowledge base (including parameters of membership functions and set of fuzzy rules), therefore the main problem of designing ICS is building optimal robust KB, which guarantee high control quality in the presence of the abovementioned control difficulties in any complex dynamic systems.

Experts for creation KB ICS are sometimes used, and this design methodology able to achieve control goals, but not always. Even experienced expert have difficulties to

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 24–36, 2011.

find an optimal KB[1] of fuzzy controller (FC) in situations of controlling nonlinear CO with stochastic noises.

Development of FC is one of the most perspective areas of fuzzy systems. For CO developers, fuzzy systems are so attractive because of the fact that they are universal "approximator" systems with poorly known dynamics and structure. In addition, they allow you to control dynamic object without expert.

2 Design Technology Knowledge Bases on Soft Computing

Application of Fuzzy Neural Networks cannot guarantee to achieve the required accuracy of approximation of the teaching signal (TS), received by genetic algorithm (GA). As a result, an essential change in external conditions is a loss of accuracy to achieve the control goal. However decision of this problem can solve by new developed tool Soft Computing Optimizer (SCO) [1, 2]. Using the design technology by SCO and previously received TS, describing the specific situation of control, it is possible to design a robust KB for control complex dynamic CO. The benchmarks of variety CO and control systems based on this approach can be found in [3].

The designed (in the general form for random conditions) robust FC for dynamic CO based on the KB optimizer with the use of soft computing technology (stage 1 of the information design technology - IDT) can operate efficiently only for fixed (or weakly varying) descriptions of the external environment. This is caused by possible loss of the robustness property under a sharp change of the functioning conditions of CO's: the internal structure of CO's, control actions (reference signal), the presence of a time delay in the measurement and control channels, under variation of conditions of functioning in the external environment, and the introduction of other weakly formalized factors in the control strategy.

To control dynamical object in different situations one has to consider all of them, i.e. design the required number of KB, the use of which will be achieved the required level of robustness control. But how can you determine what KB has to be used in the current time?

A particular solution of a given problem is obtained by introducing a generalization of strategies in models of fuzzy inference on a finite set of FC's designed in advance in the form of new quantum fuzzy inference (QFI) [4].

3 ICS Model Based on Quantum Fuzzy Inference

In the proposed model of the quantum algorithm for QFI the following actions are realized [5]:

[1] Optimal base is called base this optimal parameters of membership functions and numbers of rule, according to approximation with required accuracy of the optimal control signal.

1. The results of fuzzy inference are processed for each independent FC;
2. Based on the methods of quantum information theory, valuable quantum information hidden in independent (individual) knowledge bases is extracted;
3. In on-line, the generalized output robust control signal is designed in all sets of knowledge bases of the fuzzy controller.

In this case, the output signal of QFI in on-line is an optimal signal of control of the variation of the gains of the PID controller, which involves the necessary (best) qualitative characteristics of the output control signals of each of the fuzzy controllers, thus implementing the self-organization principle.

Therefore, the domain of efficient functioning of the structure of the intelligent control system can be essentially extended by including robustness, which is a very important characteristic of control quality. The robustness of the control signal is the ground for maintaining the reliability and accuracy of control under uncertainty conditions of information or a weakly formalized description of functioning conditions and/or control goals.

QFI model based on physical laws of quantum information theory, for computing use unitary invertible (quantum) operators and they have the following names: superposition, quantum correlation (entangled operators), interference. The forth operator, measurement of result quantum computation is irreversible.

In the general form, the model of quantum computing comprises the following five stages:

- preparation of the initial (classical or quantum) state $|\psi_{out}\rangle$;
- execution of the Hadamard transform for the initial state in order to prepare the superposition state [1];
- application of the entangled operator or the quantum correlation operator (quantum oracle) to the superposition state;
- application of the interference operator;
- application of the measurement operator to the result of quantum computing $|\psi_{out}\rangle$.

On Fig.1 is shown the functional structure of QFI.

This QFI model solves the problem robust control essentially-nonlinear unstable CO in unpredicted control situations, by extracting additional information from designed individual KB FC, created for different control situations, based on different optimization criteria.

Thus, the quantum algorithm in the model of quantum fuzzy inference is a physical prototype of production rules, implements a virtual robust knowledge base for a fuzzy PID controller in a program way (for the current unpredicted control situation), and is a problem-independent toolkit [10].

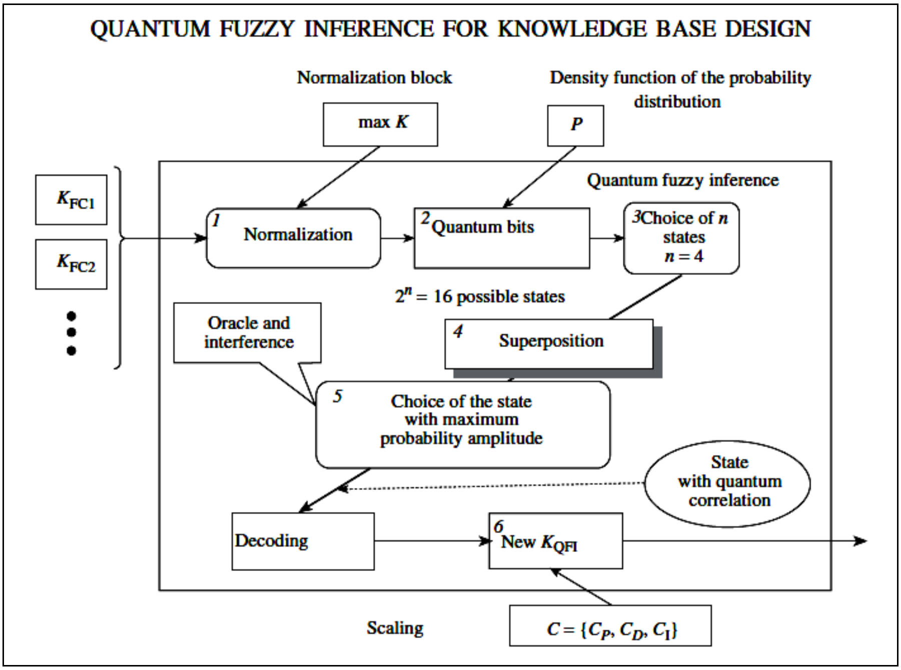

QUANTUM FUZZY INFERENCE FOR KNOWLEDGE BASE DESIGN
Normalization block
Density function of the probability distribution
max K
P
Quantum fuzzy inference
K_{FC1}
K_{FC2}
1 Normalization
2 Quantum bits
3 Choice of n states $n = 4$
$2^n = 16$ possible states
Oracle and interference
4 Superposition
5 Choice of the state with maximum probability amplitude
State with quantum correlation
Decoding
6 New K_{QFI}
Scaling
$C = \{C_P, C_D, C_I\}$

Fig. 1. The functional structure of QFI in real time

On Fig. 2 is shown intelligent robust control system of essentially nonlinear CO's.

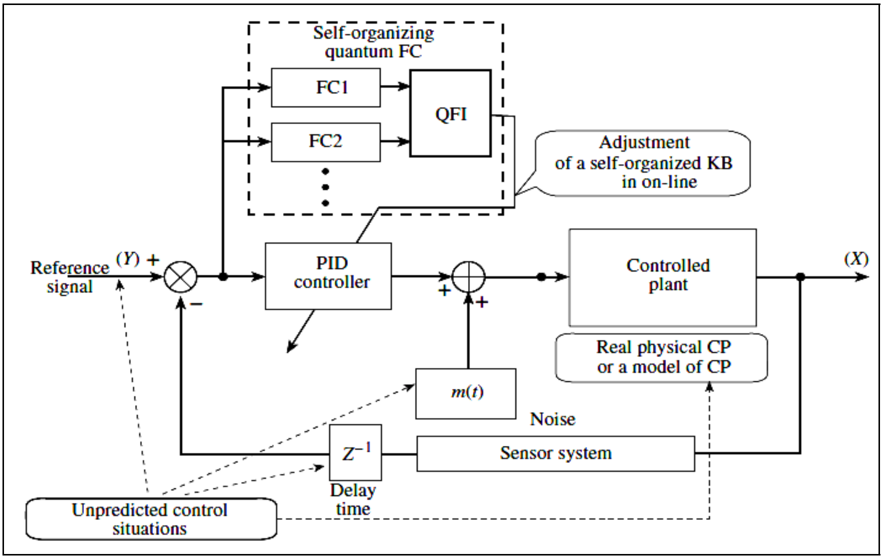

Self-organizing quantum FC
FC1
FC2
QFI
Adjustment of a self-organized KB in on-line
Reference signal
(Y) +
−
PID controller
+
+
Controlled plant
(X)
Real physical CP or a model of CP
m(t)
Noise
Z⁻¹
Sensor system
Delay time
Unpredicted control situations

Fig. 2. Principle structure of a self-organizing ICS in unpredicted control situations

The next part of this article will describe the benchmark and the results of simulations by using developed design technology of ICS.

4 Simulation Results of Control Object with Partial Unstable General Coordinates

As Benchmark example we choose the popular “Swing” dynamic system. Dynamic peculiarity of this system is consisted in following: one generalized coordinate is local unstable (angle) and another coordinate is global unstable (length). Model of “swing” dynamic system (as dynamic system with globally and locally unstable behavior) is shown on Fig.3.

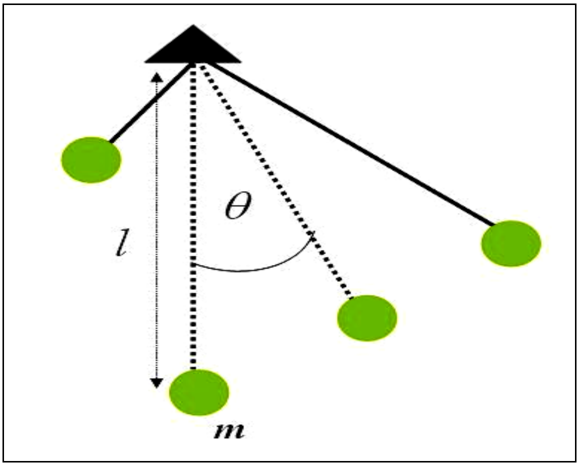


Fig. 3. Swing dynamic system

Swing dynamic system behavior under control is described by second-order differential equations for calculating the force to be used for moving a pendulum:

$$\begin{aligned} &\ddot{x}+(2\frac{\dot{y}}{y}+\frac{c}{my^{2}})\dot{x}+\frac{g}{y}\sin x=u_{1}+\xi_{1}(t) \\ &\ddot{y}+2k\dot{y}-y\dot{x}^{2}-g\cos x=\frac{1}{m}(u_{2}+\xi_{2}(t)). \end{aligned} \tag{1}$$

Equations of entropy production rate are the following:

$$\frac{dS_{\theta}}{dt}=2\frac{\dot{l}}{l}\dot{\theta}\cdot\dot{\theta};\quad \frac{dS_{l}}{dt}=2k\,\dot{l}\cdot\dot{l}. \tag{2}$$

Swing motion, described by Eqs (1), (2), show that a swing system is the globally unstable along generalized coordinate l and locally unstable along generalized coordinate θ. Also model (1) has nonlinear cross links, affecting to local unstable by generalized coordinate x.

In Eqs (1), (2) x and y — generalized coordinates; g — acceleration of gravity, m — pendulum weight, l — pendulum length, k — elastic force, c — friction coefficient, $\xi(t)$ — external stochastic noise, u_1 and u_2 — control forces. Dynamic behavior of swing system (free motion and PID control) is demonstrated on Fig 4.

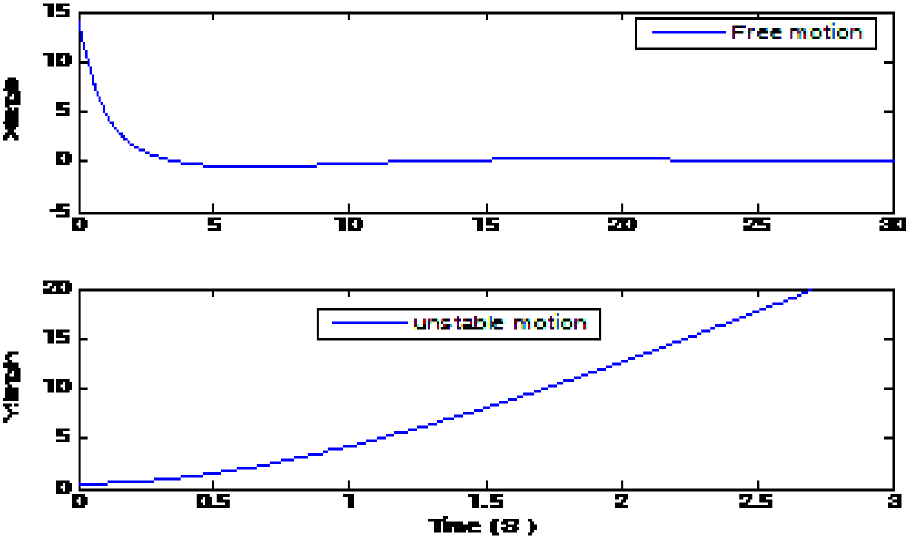


Fig. 4. Swing system free motion

Control problem: design a smart control system to move the swing system to the given angle (reference x) with the given length (reference y) in the presence of stochastic external noises and limitation on control force.

Swing system can be considered as a simple prototype of a hybrid system consisting of a few controllers where a problem of how to organize a coordination process between controllers is open (problem of coordination control).

Control task: Design robust knowledge base for fuzzy PID controllers capable to work in unpredicted control situations.

Consider excited motion of the given dynamic system under two fuzzy PID-control and design two knowledge bases for giving teaching situation (Table1).

Table 1. Teaching control situation

Teaching situations:
Noise x: Gaussian (max amplitude = 1);
Noise y: Gaussian (max amplitude = 2);
Sensor's delay time_x = 0.001 s;
Sensor's delay time_y= 0.001s;
Reference signal_x = 0; Reference signal_y = 2;
Model parameters = (kmc)=(0.4 0.5 2)
Control force boundaries: $\lvert U_x \rvert \leq 10(N)$, $\lvert U_y \rvert \leq 10(N)$

Investigate robustness of three types of spatial, temporal and spatiotemporal QFI correlations and choose best type of QFI for the given control object and given teaching conditions.

On Figs 5, 6 comparisons of three quantum fuzzy controllers (QFC) control performance based on three types of QFI (spatial, temporal and spatiotemporal QFI correlations) are shown for the teaching situation.

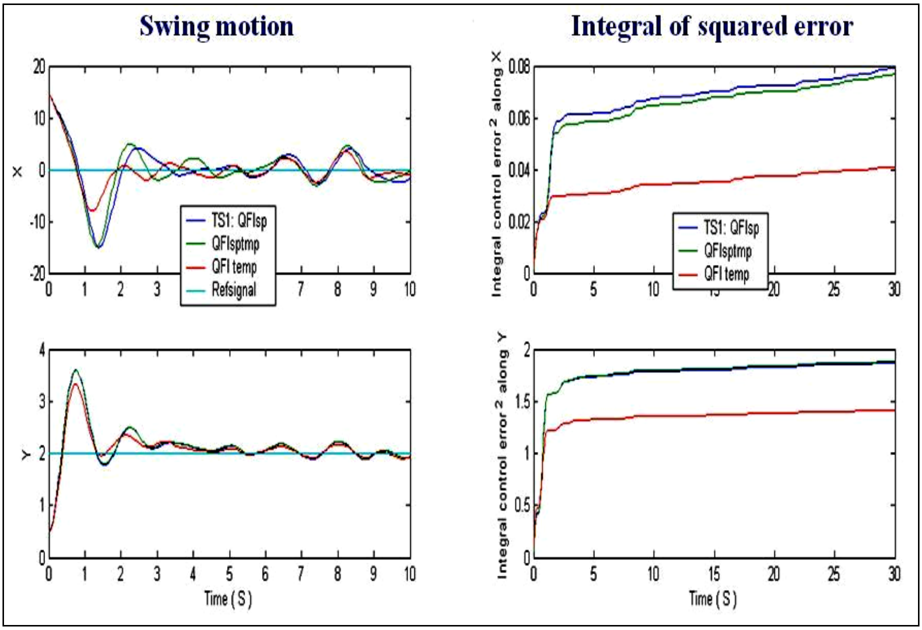


Fig. 5. Comparison of three types of quantum correlations

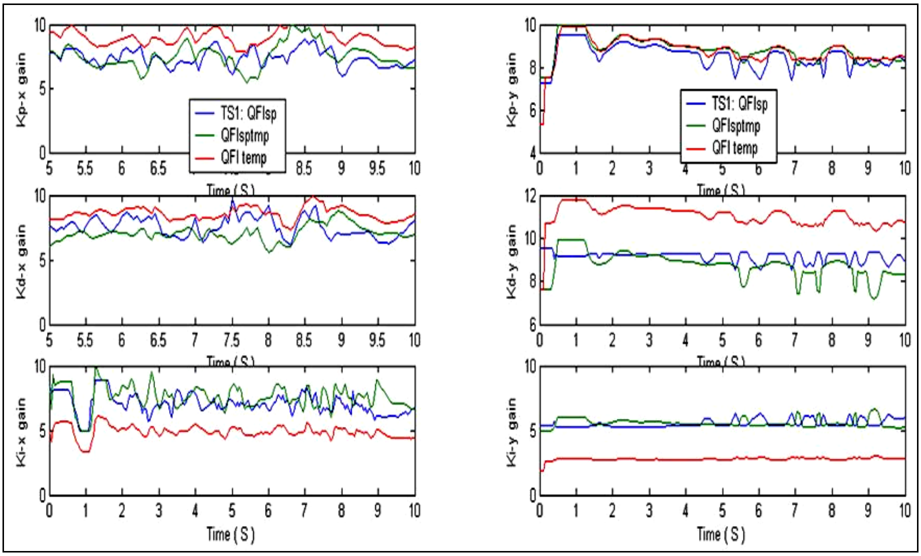


Fig. 6. Control laws comparison

Temporal QFI is better from minimum control error criterion. Choose temporal QFI for further investigations of robustness property of QFI process by using modelled unpredicted control situations.

Consider comparison of dynamic and thermodynamic behavior of our control object under different types of control: FC1, FC2, and QFC (temporal).

Comparison of FC1, FC2 and QFC performances is shown on Figs 7, 8.

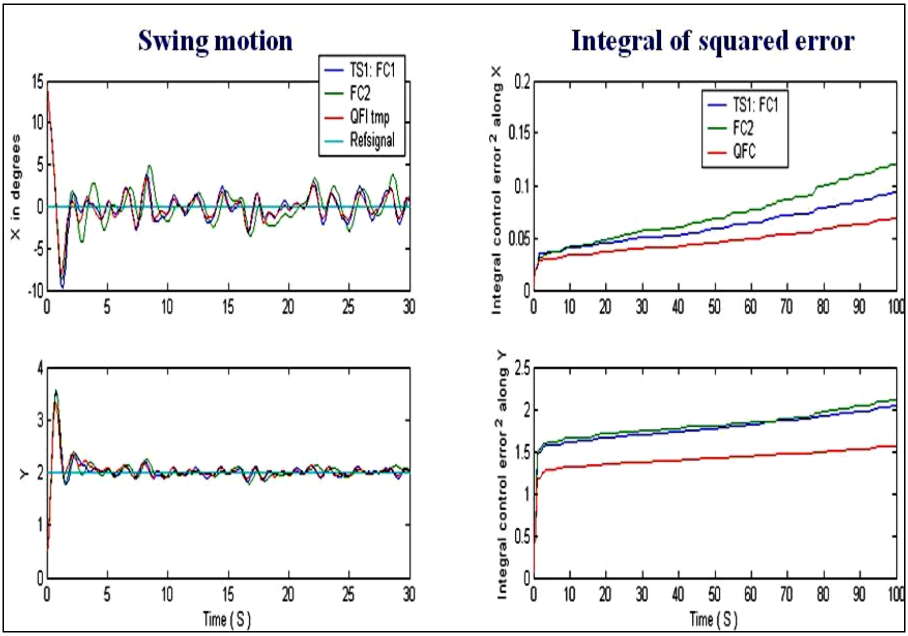


Fig. 7. Swing motion and integral control error comparison in TS situation

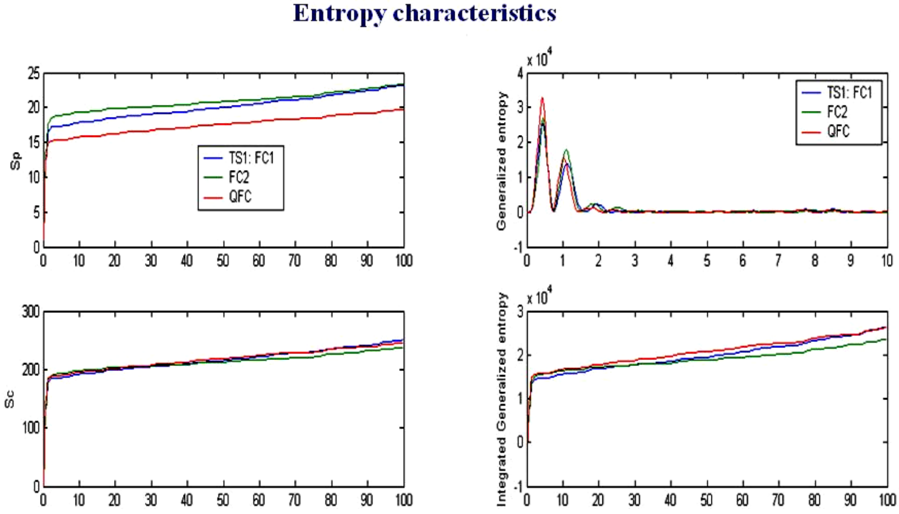


Fig. 8. Comparison of entropy production in control object (Sp) and in controllers (left) and comparison of generalized entropy production (right)

From the minimum control error criterion in teaching condition QFC has better performance than FC1, FC2.

Consider now behavior of our control object in unpredicted control situations and investigate robustness property of designed controllers (Table 2).

Table 2. Unpredicted control situations

Unpredicted situation 1:	**Unpredicted situation 2:**
Noise x: **Gaussian** (max = 1);	Noise x: **Rayleigh**(max = 1);
Noise y:**Gaussian** (max = 2);	Noise y: **Rayleigh**(max = 2);
Sensor's delay time_x= 0.008 s;	Sensor's delay time_x= 0.001 s;
Sensor's delay time_y= 0.008s;	Sensor's delay time_y = 0.001s;
Reference signal_x = 0;	Reference signal_x = 0
Reference signal_y = 2;	Reference signal_y= 2;
Model parameters:	Model parameters :
(kmc)=(0.4 0.5 2)	(kmc)=(0.4 0.5 2)
Control force boundaries:	Control force boundaries:
$\|U_x\| \le 10(N), \quad \|U_y\| \le 10(N)$	$\|U_x\| \le 10(N), \quad \|U_y\| \le 10(N)$

Unpredicted situation 1 Comparison of FC1, FC2 and QFC performances in situation 1 (see, Figs 9 – 11).

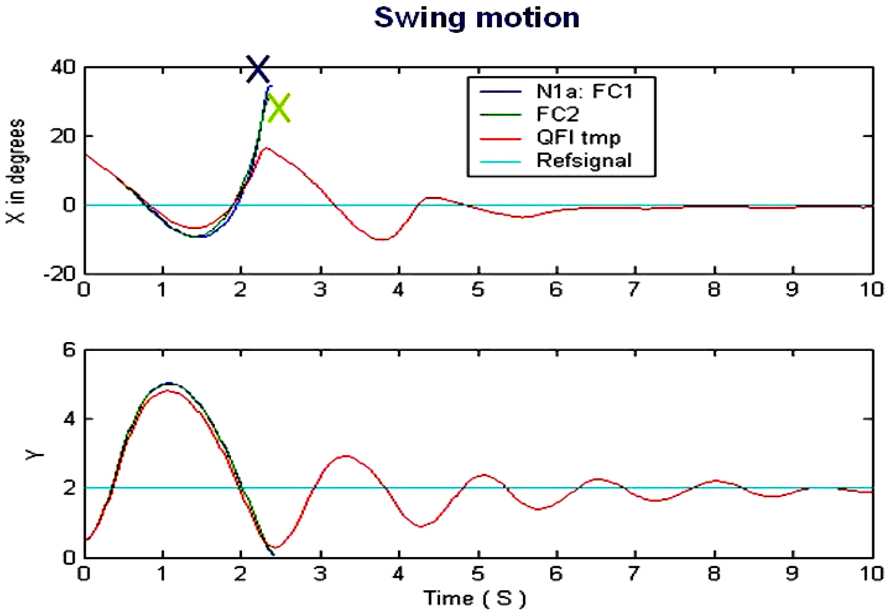


Fig. 9. Swing motion and integral control error comparison in unpredicted control situation 1

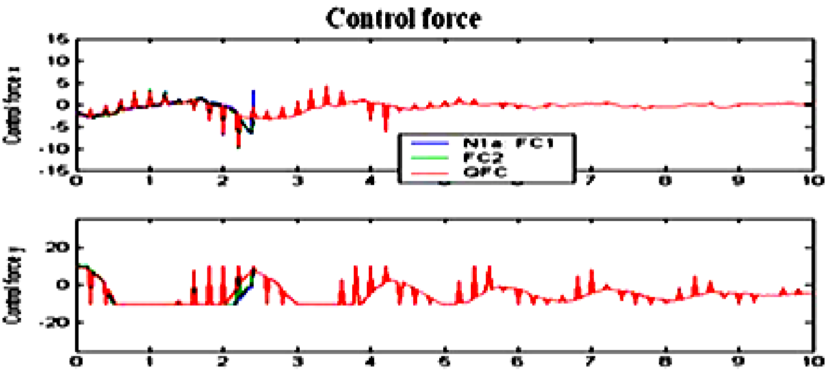


Fig. 10. Control forces comparison in unpredicted control situation 1

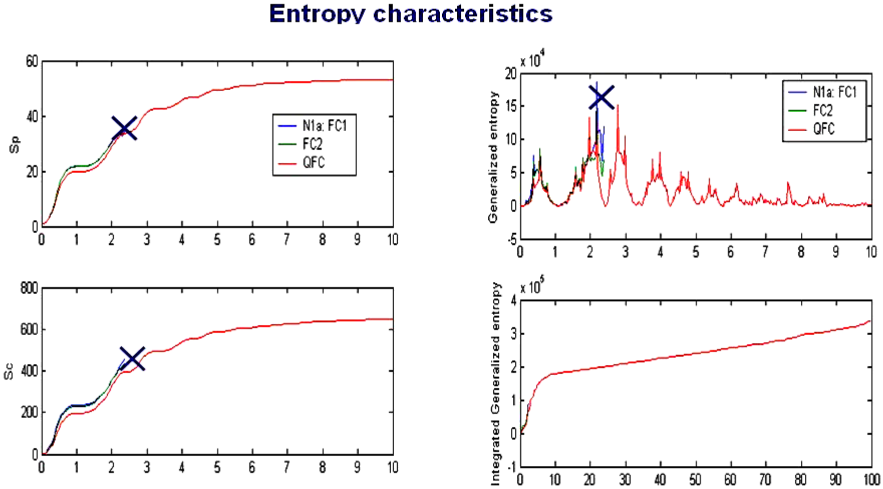


Fig. 11. Comparison of entropy production in control object (Sp) and in controllers (left) and comparison of generalized entropy production (right) in unpredicted control situation 1

FC1 and FC2 controllers are failed in situation 1. QFC is robust.

Unpredicted situation 2 Comparison of FC1, FC2 and QFC performances in situation 2 (see, Figs 12– 14).

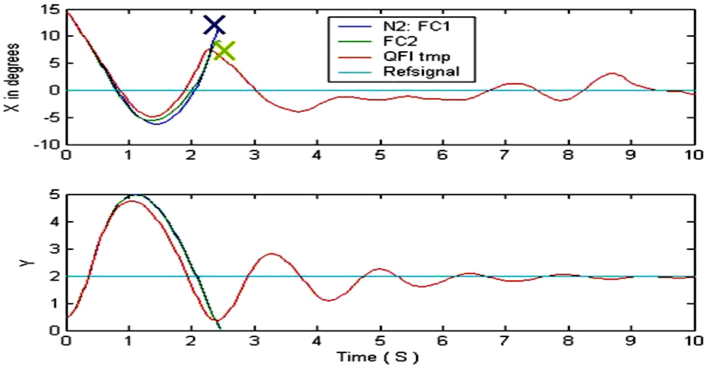


Fig. 12. Swing motion and integral control error comparison in unpredicted control situation 2

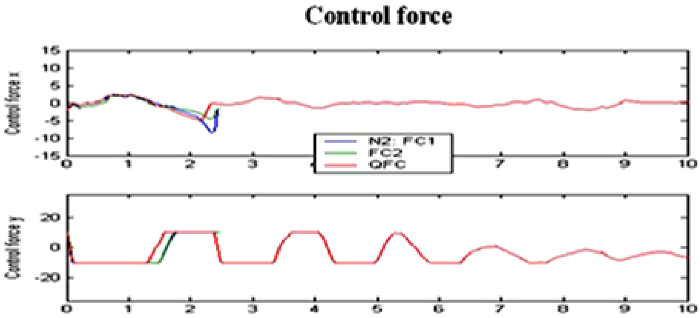


Fig. 13. Control forces comparison in unpredicted control situation 2

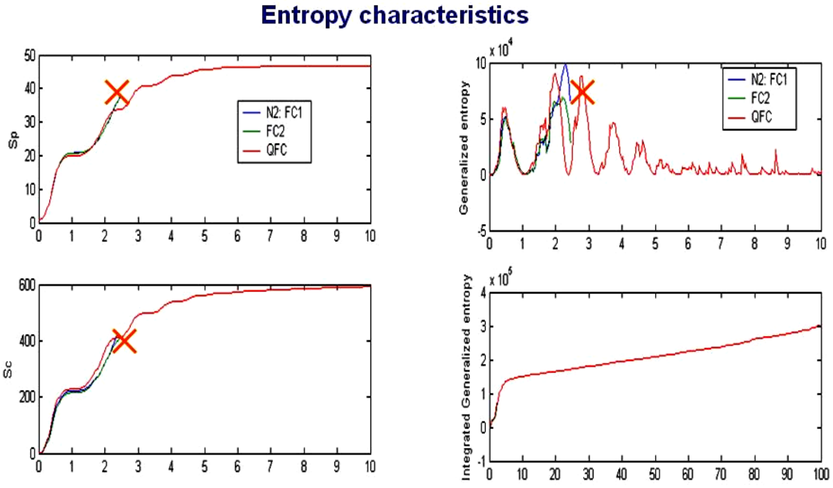


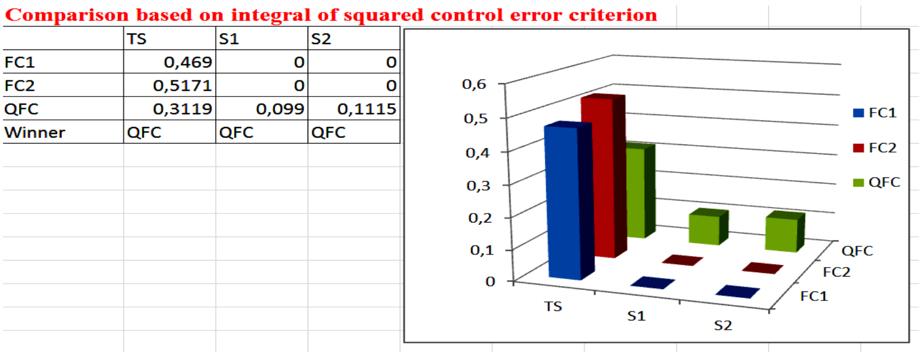
Fig. 14. Comparison of entropy production in control object (Sp) and in controllers (left) and comparison of generalized entropy production (right) in unpredicted control situation 2

FC1 and FC2 controllers are failed in situation 2. QFC is robust.

5 General Comparison of Control Quality of Designed Controllers

Consider now general comparison of control quality of four designed controllers (FC1, FC2, QFC based on temporal QFI with 2 KB). We will use the control quality criteria of two types: dynamic behavior performance level and control performance level.

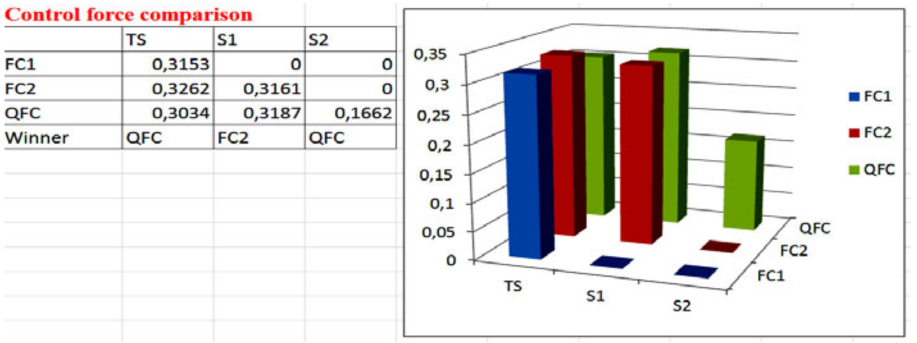
Control quality comparison is shown on Figs below15, 16.



Comparison based on integral of squared control error criterion

	TS	S1	S2
FC1	0,469	0	0
FC2	0,5171	0	0
QFC	0,3119	0,099	0,1115
Winner	QFC	QFC	QFC

Fig. 15. Comparison based on integral of squared control error criterion



Control force comparison

	TS	S1	S2
FC1	0,3153	0	0
FC2	0,3262	0,3161	0
QFC	0,3034	0,3187	0,1662
Winner	QFC	FC2	QFC

Fig. 16. Comparison based on simplicity of control force

- QFCis robust in all situations;
- FC1 controller is not robust in**2, 3** situations;
- FC2 controller is not robust in **2, 3** situations.

Thus, ICS with QFI based on two KB and temporal correlation type has the highest robustness level (among designed controllers) and show the highest self-organization degree.

From simulation results follows an unexpected (for the classical logic and the methodology of ICS design) conclusion: with the help of QFI from two not robust (in unpredictable situation) controllers (FC1 and FC2) one can get robust FC online.

6 Conclusions

In this article modeling behavior CO has been made (pendulum with variable length) based on QFI. The obtained simulation results show that designed KB of FC is robust in terms of criteria for control quality such as minimum error control and entropy production, as well as the minimum applied control force. Presented design technology allows achieving control goal even in unpredicted control situations.

References

1. Litvintseva, L.V., Ulyanov, S.S., Takahashi, K., et al.: Intelligent robust control design based on new types of computation. Pt 1. In: New Soft Computing Technology of KB-Design of Smart Control Simulation for Nonlinear Dynamic Systems, vol. 60, Note del Polo (Ricerca), Universita degli Studi di Milano, Milan (2004)
2. Litvintseva, L.V., Ulyanov, S.V., et al.: Soft computing optimizer for intelligent control systems design: the structure and applications. J. Systemics, Cybernetics and Informatics (USA) 1, 1–5 (2003)
3. Litvintzeva, L.V., Takahashi, K., Ulyanov, I.S., Ulyanov, S.S.: Intelligent Robust control design based on new types of computations, part I. In: New Soft Computing Technology of KB-Design Benchmarks of Smart Control Simulation for Nonlinear Dynamic Systems, Universita degli Studi di Milano, Crema (2004)
4. Litvintseva, L.V., Ulyanov, I.S., Ulyanov, S.V., Ulyanov, S.S.: Quantum fuzzy inference for knowledge base design in robust intelligent controllers. J. of Computer and Systems Sciences Intern. 46(6), 908–961 (2007)
5. Ulyanov, S.V., Litvintseva, L.V.: Design of self-organized intelligent control systems based on quantum fuzzy inference: Intelligent system of systems engineering approach. In: Proc. of IEEE Internat. Conf. On Systems, Man and Cybernetics (SMC 2005), Hawaii, USA, vol. 4 (2005)
6. Ulyanov, S.V., Litvintseva, L.V., Ulyanov, S.S., et al.: Self-organization principle and robust wise control design based on quantum fuzzy inference. In: Proc. of Internat. Conf. ICSCCW 2005, Antalya. Turkey (2005)
7. Litvintseva, L.V., Ulyanov, S.V., Takahashi, K., et al.: Design of self-organized robust wise control systems based on quantum fuzzy inference. In: Proc. of World Automation Congress (WAC 2006): Soft Computing with Industrial Applications (ISSCI 2006), Budapest, Hungary, vol. 5 (2006)
8. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge Univ. Press, UK (2000)
9. Ulyanov, S.V.: System and method for control using quantum soft computing. US patent. — No. 6,578,018B1 (2003)
10. Ulyanov, S.V., Litvintseva, L.V., Ulyanov, S.S., et al.: Quantum information and quantum computational intelligence: Backgrounds and applied toolkit of information design technologies, vol. 78–86. Note del Polo (Ricerca), Universita degli Studi di Milano, Milan (2005)

Type-2 Neuro-Fuzzy Modeling for a Batch Biotechnological Process

Pablo Hernández Torres[1], María Angélica Espejel Rivera[2],
Luis Enrique Ramos Velasco[1,3], Julio Cesar Ramos Fernández[3],
and Julio Waissman Vilanova[4]

[1] Centro de Investigación en Tecnologías de Información y Sistemas,
Universidad Autónoma del Estado Hidalgo, Pachuca de Soto, Hidalgo, México, 42090
[2] Universidad la Salle Pachuca, Campus La Concepción, Av. San Juan Bautista de La Salle No. 1. San Juan Tilcuautla, San Agustín Tlaxiaca, Hgo. C.P. 42160. Pachuca, Hidalgo. México
[3] Universidad Politécnica de Pachuca, Carretera Pachuca-Cd. Sahagún, Km. 20, Rancho Luna, Ex-Hacienda de Sta. Bárbara, Municipio de Zempoala, Hidalgo, México
[4] Universidad de Sonora, Blvd. Encinas esquina con Rosales s/n C.P. 83000,
Hermosillo, Sonora, México
juliowaissman@mat.uson.mx

Abstract. In this paper we developed a Type-2 Fuzzy Logic System (T2FLS) in order to model a batch biotechnological process. Type-2 fuzzy logic systems are suitable to drive uncertainty like that arising from process measurements. The developed model is contrasted with an usual type-1 fuzzy model driven by the same uncertain data. Model development is conducted, mainly, by experimental data which is comprised by thirteen data sets obtained from different performances of the process, each data set presents a different level of uncertainty. Parameters from models are tuned with gradient-descent rule, a technique from neural networks field.

1 Introduction

Biological processes are the most common technology for waste water treatment due its comparative low cost and efficiency, however this kind of systems are complex because its strong nonlinearities, unpredictable disturbances, behavior's poor and incomplete understanding, time-variant characteristics and uncertainties [1].

These reasons make suitable the use of alternative modeling techniques, beyond the classical first-order nonlinear differential equations as are usually employed, to understand and explain this kind of processes which is necessary for control and optimization purposes.

Biodegradation of toxic compounds carried out in bioreactors under batch operation (like a SBR) are controlled and optimized through its principal variables, the initial substrate and biomass concentrations, S_0 and X_0 respectively, in the filling cycle. Moreover, this type of operation in reactors gives us different performances or biodegradation patterns for different initial relations of those variables.

TYPE-2 fuzzy sets (T2 FSs), originally introduced by Zadeh , provide additional design degrees of freedom in Mamdani and TSK fuzzy logic systems (FLSs), which

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 37–45, 2011.

can be very useful when such systems are used in situations where lots of uncertainties are presented [2].

Fuzzy logic works with vagueness in classes or sets defined in an universe of discourse in that a sense we can not establish if a element from the universe belongs to a class or not, but actually we can say such element belongs to all classes in a certain degree: zero for absolute no membership and one for complete membership. Type-2 fuzzy logic add more membership degrees to elements and furthermore assign to those degrees a certainty grade or weight; higher types of fuzzy logic add certainty grades to certainty grades and the like [3,4,5,6,7]. So, usual FISs (Fuzzy Inference Systems) are suitable for linguistic representations of processes and higher types FISs for modeling, for example, with uncertain data and non clear membership functions [8,9,10]; moreover, uncertain data can be used to modelling with fuzzy numbers as well [11].

Interval type-2 FLSs provide a way to handle knowledge uncertainty, data mining and knowledge discovery are important research topics that are being studied by researchers of neural networks, fuzzy logic systems, evolutionary computing, soft computing, artificial intelligence, etc [9,12]. Deriving the analytical structure of a fuzzy controller with the product AND operator is relatively simple; however, a fuzzy controller involving the other operator is far more difficult. Structurally, a T2 fuzzy controller is more complicated than its T1 counterpart as the former has more components (e.g., type reducer), more parameters (e.g., T2 fuzzy sets), and a more complex inference mechanism [13].

We believe that interval type-2 FLSs have the potential to solve data mining and knowledge discovery problems in the presence of uncertainty.

This article is organized as follows: After a brief description of the data and substate model in Section 2, experimental results are shown in Section 3 and Section 4, followed by the conclusions in Section 6.

2 Data and Substrate Model

Beyond the substrate and biomass, intermediate product concentration (I) of microorganisms' measurements are part of the data sets, this variable is important as it causes inhibition in the consumption activity of biomass [14].

The discrete nonlinear first-order ordinary differential equation (1) was enough with one set of parameters, see Table 1, but when we have thirteen data sets obtained from different performances of the process which is our case where each data set presents a different level of uncertainty to be modeled, however for intermediate concentration it was not possible and thus the necessity for using a fuzzy model. Figure 1 layouts the estimations from the substrate model.

Model development is conducted, mainly, by experimental data which is comprised by

$$S(k+1) = S(k) - .001T\frac{q_{S_{\max}} S(k)}{K_{\mathrm{S}} + S(k) + S(k)^n / K_{\mathrm{i}}} \tag{1}$$

As cell decay is negligible and cell growth is slow and quasi-constant over several bioreactor cycles, it is considered constant and thus S and I dynamics are unaffected by the X ones.

Table 1. Coefficients set for the discrete nonlinear ODE that worked for all substrate biodegradation patterns

Kinetic constant	Symbol	Value
substrate consumption specific rate	$q_{S_{\text{max}}}$	29.7 $mg/gMES$ per h
half-saturation constant	K_{S}	77.5 mg/l
inhibition constant	K_{i}	738.61 mg/l
a constant	n	2.276

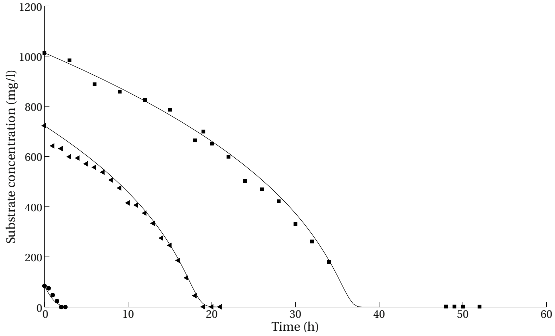


Fig. 1. Measured substrate for different biodegradation patterns corresponding to different $S(0)$: ($\bullet$) 84.05 mg/l, ($\triangleright$) 722.74 mg/l and ($\diamond$) 1013.15 mg/l. Solid line shows the simulated model.

Fig. 2 shows the measurements from different data sets, as can be seen the intermediate presents two phases, one of production and another of consumption, where the division of both is just in the point where substrate has been exhausted indicating that once it has happened microorganisms start to feed of intermediate.

3 Type-1 Neuro-Fuzzy Model

3.1 Model Structure

Regression models are adequate to model time series of linear and nonlinear systems [15], so as the model is a nonlinear and first-order one a NARX (Nonlinear AutoRegressive with eXogenous input) regression structure was proposed with time delays $n_u,\ n_{y^*} = 1$, a representation of such regression structure is given by

$$y(k+1) = F(y^*(k), ..., y^*(k - n_{y^*} + 1), u(k), ..., u(k - n_u + 1)), \qquad (2)$$

where F is the true relation between the involved variables which will be approximated by the fuzzy system f, the inputs u of the model are chosen to be S and $S(0)$ whereas I will be the model output, therefore we have that $I(k+1) = F(S(k), I^*(k), S(0))$.

The used fuzzy system f, either in type-1 or type-2 fashion, is a TS (Takagi-Sugeno) fuzzy logic system (FLS). It is considered an universal approximator [16] and globally represents the non-linear relationship F but whose rules are local linear models which relates input variables to the output one with a linear version of (2); then rules from f are as follows

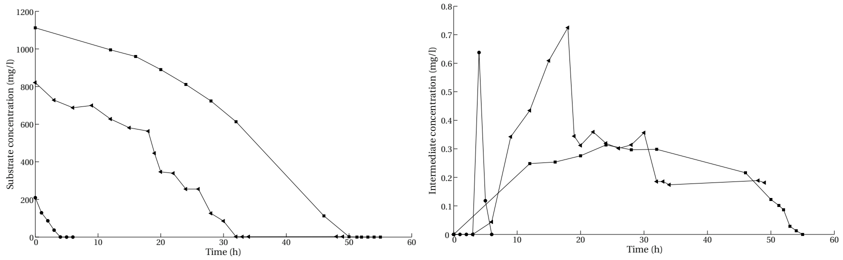


Fig. 2. Several data sets or batches showing substrate consumption (left) and intermediate production (right) for different S_0 values:(●) 209.39 mg/l, (▷) 821.02 mg/l and (◇) 1013.15 mg/l

$$\mathcal{R}_i : \textbf{IF } S(k) \textbf{ is } A_{i,1} \textbf{ and } I^*(k) \textbf{ is } A_{i,2} \textbf{ and } S(0) \textbf{ is } A_{i,3} \textbf{ THEN} \tag{3}$$

$$I_i(k+1) = a_{i,1}S(k) + a_{i,2}I^*(k) + a_{i,3}S(0) + b_i, \tag{4}$$

where the fuzzy sets $A_{i,j}$ are represented by the Gaussian MFs (Membership Functions) $\mu_{A_{i,j}}$. A Gaussian MF is easily derivable which is useful when gradient techniques are used.

Gradient descent formula denoted by

$$\boldsymbol{\omega}(n+1) = \boldsymbol{\omega}(n) - \alpha(n)\nabla J(\boldsymbol{\omega}(n)) \tag{5}$$

let us to find the rules parameters optimal values [17] by minimizing an error measure function J that is commonly defined by

$$J = \sum_{k=1}^{N} e_k \quad \text{with} \quad e_k = \frac{1}{2}\left(y_k - y_k^*\right)^2, \tag{6}$$

where y_k is the estimated output of the model, in our case I, and y^* is the desired output, I^*, in the sampling instant k.

Gradient $\nabla J(\boldsymbol{\omega}(k))$ points out to the minimum of (6) according to parameters vector $\boldsymbol{\omega}$ which have the antecedent and consequent MF parameters of (3). Although all parameters could be found with the gradient learning rules, an hybrid method is mostly used because due it computes the consequent parameters with least-squares [18] it avoids local minima and has a faster convergence. As can be seen input and output data must be proportioned to tune parameters as it is needed to compute J's e_k.

Optimal number of rules was looked by a try and error method, MF coefficients was initialized with grid partition of input space, learning coefficient α from (5) and total number of epochs was specified empirically;

3.2 Model Estimates

As data samples cardinality is less, by much, than the total parameters to be estimated for the fuzzy model, data was interpolated assuring this interpolation provided lightly bigger number of data values than parameters in order to have a correct parameters estimation during training [19]. Furthermore, the interval between interpolates is the same as the sampling period T of (1).

Not more than 10 epochs of training were needed and an initial learning rate α of 0.01 with a final value of .0121 was enough together with two MFs for substrate and intermediate and five MFs for S_0 (total of 20 rules) to get the intermediate estimates.

4 Type-2 Neuro-Fuzzy Model

Same type-1 model structure applies to the type-2 model moreover the gradient descent learning rule is used in similar way to find the parameters of the fuzzy system, as is detailed in [20], however the learning rules are different and more complex because the relation between J and the parameters changes and the membership functions now are of the type-2. The Fig. 3 shows the type-2 neuro-fuzzy using in this paper.

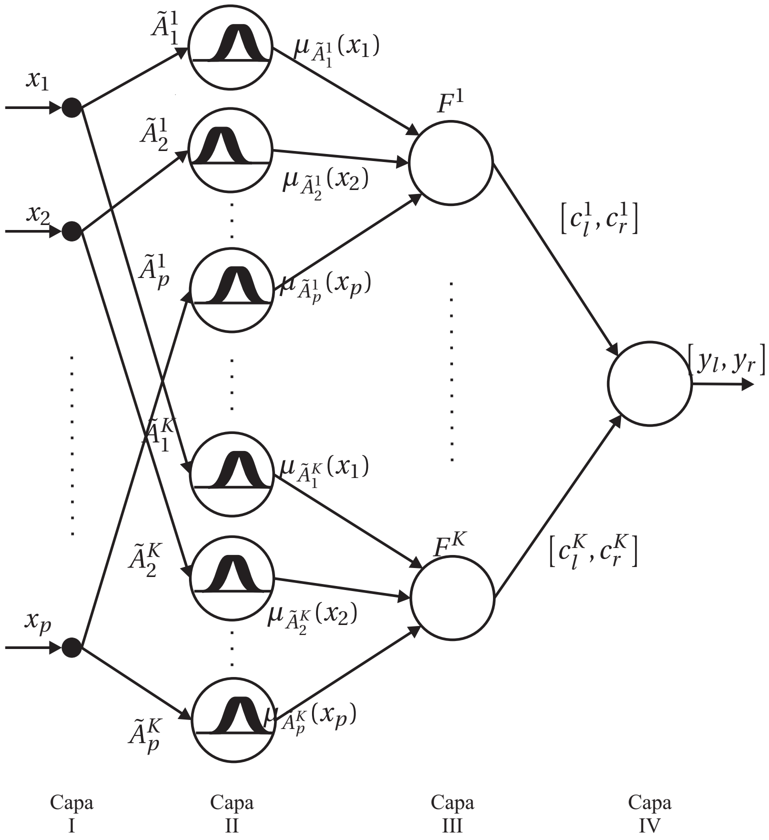


Fig. 3. Type-2 neuro-fuzzy

$$F^i(\boldsymbol{x}) = \sqcap_{j=1}^{p} \mu_{\tilde{A}_j^i}(x_j), \tag{7}$$

with $\boldsymbol{x} = [x_1, x_2, \ldots, x_p]$ is the input vector. $\sqcap$ is the meet operation for type-2 intersection fuzzy sets. If $\tilde{A}_j^i$ y C_j^i are interval sets, type-2 and type-1, respectively, then we have TS-2 model interval. A type-2 version of TK (Takagi-Sugeno) FLS ([3]) is given by

$$\mathcal{R}_i : \textbf{IF } S(k) \textbf{ is } \tilde{A}_1^i \textbf{ and } I^*(k) \textbf{ is } \tilde{A}_2^i \textbf{ and } S(0) \textbf{ is } \tilde{A}_3^i \textbf{ THEN} \tag{8}$$

$$I_i(k+1) = A_1^i S(k) + A_2^i I(k) + A_3^i S(0) + B^i, \tag{9}$$

where now type-2 fuzzy sets $\tilde{A}_j^i$ are represented by type-2 MFs $\mu_{\tilde{A}_j^i}$ and coefficients A_j^i and B^i are type-1 fuzzy sets. Now, the output of the system and of every rule is an interval set that represents the uncertainty of the process. The complexity of the system is evident just not for the increasing number of parameters but for the larger and tedious operations.

Interval MFs were used in antecedents and consequents, the reason is that even though they are less complex MFs they offer as goods and even better results than the most complex ones. Interval type-2 MF and the analog type-1 MF are shown in Fig. 4; in the antecedent, Gaussian MFs with uncertain mean were employed producing piecewise functions.

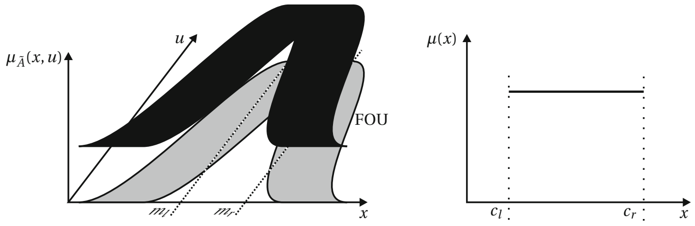


Fig. 4. Type-1 and type-2 interval MFs used for the type-2 TK fuzzy model, type-2 (left) shows the FOU (Footprint Of Uncertainty) due to the uncertain mean of Gaussian function

Another difference respect to the type-1 modeling procedure is that in this model the initial values of parameters were taken from a quasi-tuned type-1 FLS which a percentage of uncertainty was added.

Same number of rules and MFs were used for the type-2 model, however there was more interpolated data due there are more parameters with the same quantity of rules. The training was carried out with 20 epochs and $\alpha = 0.001$, more epochs and littler learning rate prevented the gradient algorithm to oscillate around the minimum of the function J.

5 Learning and Testing

We determined the learning rate in empirically form, which was different for each batch of experimental data. During the learning network were used T2FNN training and test sets for ANFIS models. Below are the results obtained.

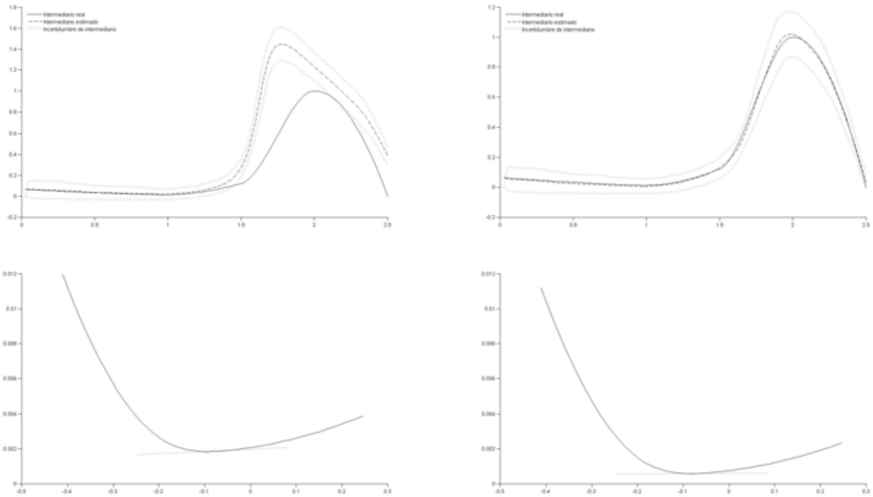


Fig. 5. Estimate (up) and function optimization (bottom) before (left) and after (right) learning

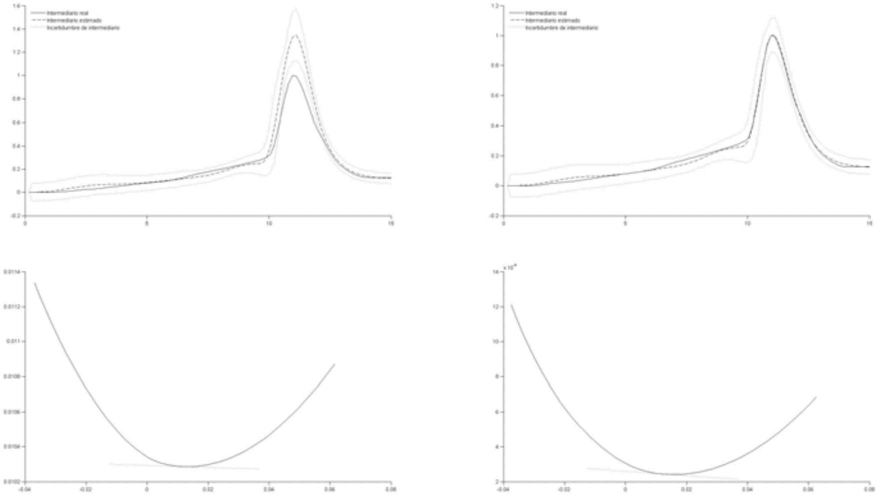


Fig. 6. Estimate (up) and function optimization (bottom) before (left) and after (right) learning

Lot of low initial concentration S_0. The Fig. 5 shows the estimation of the intermediary before and after learning the data set for $S_0 = 84.05$ milligrams. It also shows the gradient function with respect to the parameter optimization $c^1_{1,l}$.

Lot of medium initial concentration S_0. The Fig. 6 shows a simulation for a lot with initial concentration of substrate $S_0 = 432.72$ milligrams. The optimization function and gradient are represented over the same parameter as the graph above.

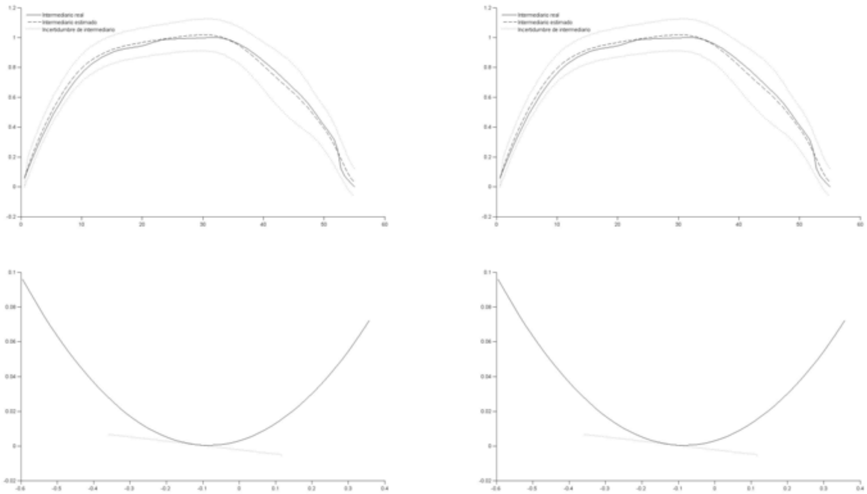


Fig. 7. Estimate (up) and function optimization (bottom) before (left) and after (right) learning

Lot of high initial concentration S_0. The Fig. 7 shows the case high initial concentration of substrate ($S_0 = 1112.21mg$), for this experiment we did not require any learning time because if it starts the learning to minimize the error of the training data on the model began to learn and error for test data began to grow. Table 2 shows as was the learning of all experimental lots that we used.

Table 2. Training variables for all data sets

S_0 (mg)	MFs for S	MFs for I	Rules	α	Epochs	RMSE initial	RMSE final
40.07	4	2	8	0.01	4	0.3158	0.0117
84.05	4	4	16	0.01	16	0.2903	0.0133
209.39	4	2	8	0.004	5	4.3081	0.0256
432.72	4	2	8	0.007	10	0.0978	0.0141
722.74	2	5	10	0.007	2	0.6313	0.0459
821.02	4	2	8	0.001	8	0.0468	0.0424
1013.15	2	3	6	0.001	19	0.0610	0.0551
1112.21	2	2	4	—	0	0.0266	0.0266

6 Conclusions

A type-2 FLS does not eliminates uncertainty but drives it from input trough the model until the output, *i.e* the output is uncertain according the input and parameters own uncertainty, but a decision about this uncertainty may be taken at the end by means of the output defuzzification. The model will exact predict the samples trace if as defuzzification technique is used the one employed in the gradient descent learning rules derivation. So, the more the uncertainty added to model's parameters the more the supported uncertainty in the inputs and of course in the output.

Acknowledgments. Author thanks Gabriela Vázquez Rodríguez by the proportioned data used in this work from the pilot SBR plant under her supervision and Julio Waissman Vilanova for his knowledge and support about the biological process' behavior and theory.

References

1. Georgieva, O., Wagenknecht, M., Hampel, R.: Takagi-Sugeno Fuzzy Model Development of Batch Biotechnological Processes. International Journal of Approximate Reasoning 26, 233–250 (2001)
2. Mendel, J.M., John, R.I., Liu, F.: Interval type-2 fuzzy logic systems made simple. IEEE Transactions on Fuzzy Systems 14(6) (December 2006)
3. Castillo, O., Melin, P.: Type-2 Fuzzy Logic: Theory and Applications. Springer, Heidelberg (2008)
4. Ramírez, C.L., Castillo, O., Melin, P., Díaz, A.R.: Simulation of the bird age-structured population growth based on an interval type-2 fuzzy cellular structure. Inf. Sci. 181(3), 519–535 (2011)
5. Castillo, O., Melin, P., Garza, A.A., Montiel, O., Sepúlveda, R.: Optimization of interval type-2 fuzzy logic controllers using evolutionary algorithms. Soft Comput. 15(6), 1145–1160 (2011)
6. Castillo, O., Aguilar, L.T., Cázarez-Castro, N.R., Cardenas, S.: Systematic design of a stable type-2 fuzzy logic controller. Appl. Soft Comput. 8(3), 1274–1279 (2008)
7. Sepúlveda, R., Castillo, O., Melin, P., Montiel, O.: An efficient computational method to implement type-2 fuzzy logic in control applications. Analysis and Design of Intelligent Systems using Soft Computing Techniques, 45–52 (2007)
8. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: introduction and new directions. Prentice-Hall (2001)
9. Liang, Q., Mendel, J.: Interval type-2 fuzzy logic systems: Theory and design. IEEE Transactions on Fuzzy Systems 8, 535–550 (2000)
10. 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)
11. Delgado, M., Verdegay, J.L., Vila, M.A.: Fuzzy Numbers, Definitions and Properties. Mathware & Soft Computing (1), 31–43 (1994)
12. Castro, J.R., Castillo, O., Melin, P., Díaz, A.R.: A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Inf. Sci. 179(13), 2175–2193 (2009)
13. Du, X., Ying, H.: Derivation and analysis of the analytical structures of the interval type-2 fuzzy-pi and pd controllers. IEEE Transactions on Fuzzy Systems 18(4) (August 2010)
14. Vázquez-Rodríguez, G., Youssef, C.B., Waissman-Vilanova, J.: Two-step Modeling of the Biodegradation of Phenol by an Acclimated Activated Sludge. Chemical Engineering Journal 117, 245–252 (2006)
15. Ljung, L.: System Identification: Theory for the User. Prentice-Hall (1987)
16. Tanaka, K., Wang, H.O.: Fuzzy Control Systems Design and Analysis. Wiley-Interscience (2001)
17. Babuška, R., Verbruggen, H.: Neuro-Fuzzy Methods for Nonlinear System Identification. Annual Reviews in Control 27, 73–85 (2003)
18. Jang, J.S.R.: Anfis: Adaptive-network-based fuzzy inference systems. IEEE Transactions on Systems, Man, and Cybernetics 23(3), 665–685 (1993)
19. Haykin, S.: Neural Networks: a comprehensive foundation. 2nd edn. Prentice-Hall (1999)
20. Hagras, H.: Comments on "Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN)". IEEE Transactions on Systems, Man, and Cybernetics 36(5), 1206–1209 (2006)

Assessment of Uncertainty in the Projective Tree Test Using an ANFIS Learning Approach

Luis G. Martínez, Juan R. Castro, Guillermo Licea, and Antonio Rodríguez-Díaz

Universidad Autónoma de Baja California
Calzada Tecnológico 14418, Tijuana, México 22300
{luisgmo,jrcastro,glicea,ardiaz}@uabc.edu.mx

Abstract. In psychology projective tests are interpretative and subjective obtaining results based on the eye of the beholder, they are widely used because they yield rich and unique data and are very useful. Because measurement of drawing attributes have a degree of uncertainty it is possible to explore a fuzzy model approach to better assess interpretative results. This paper presents a study of the tree projective test applied in software development teams as part of RAMSET's (Role Assignment Methodology for Software Engineering Teams) methodology to assign specific roles to work in the team; using a Takagi-Sugeno-Kang (TSK) Fuzzy Inference System (FIS) and also training data applying an ANFIS model to our case studies we have obtained an application that can help in role assignment decision process recommending best suited roles for performance in software engineering teams.

Keywords: Fuzzy Logic, Uncertainty, Software Engineering, Psychometrics.

1 Introduction

Handling imprecision and uncertainty in software development has been researched [1] mainly in effort prediction, estimation, effectiveness and robustness, but never until recently in role assignment. The output of decision making process is either yes or no in two-valued logic system. The Maxim of Uncertainty in Software Engineering (MUSE) states that uncertainty is inherent and inevitable in software development processes and products [2]. It is a general and abstract statement applicable to many facets of software engineering.

Industrial and Organizational psychologists that work in personnel selection choose selection methods which are most likely correlated with performance for a specific job as Bobrow [3] has analyzed. Multiple assessment methods are used in a selection system, because one technique does not cover all knowledge, skills, abilities, and personal attributes (KSAPs) for a specific job. He has correlated selection tools with job performance, obtaining that cognitive ability and work sample tests are better predictors of job performance than measures of personality. However, it should be noted that the advent of the Big 5 factor model [4] and the development of non-clinical personality instruments, has led to a renaissance of the use of personality tests as selection techniques.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 46–57, 2011.

Effective use of psychometric instruments add value to organizations, these are used in selection and structured interview process to select more accurately people who will perform best in a role. Personality tests like Jung, Myers-Briggs, Big Five and projective tests like House-Tree-Person are used to know the sociopsychological characteristics and personality of individuals besides abilities for job placement and hiring and therefore to assign individuals to form a working team [5]. Personality tests are based on interpretation; therefore to tackle uncertainty we have found that using Fuzzy Logic has help us better define personality patterns thus recommend a best suited role for performance in software engineering teams.

This paper is a study focused on the Projective Tree Test used as a part of RAMSET's methodology, a personality based methodology used in software project development case studies, first using a Takagi-Sugeno-Kang (TSK) Fuzzy Inference System (FIS) and then training data using an Adaptive Network Based Fuzzy Inference System (ANFIS) model. The rest of the paper is organized as follows: section 2 is a brief background of personnel selection importance and related fuzzy logic approaches. Section 3 is a brief description of RAMSET methodology. Section 4 defines our Tree Test Fuzzy Model and section 5 our ANFIS trained model. Section 6 displays results of the projective tree test concluding in section 7 with observations for discussion.

2 Background

Personnel selection and assessment applies the measurement of individual differences to hiring of people into jobs where they are likely to succeed. Industrial and organizational psychologists who practice in this area use information about the job and the candidates to help a company determine which candidate is most qualified for the job. Dereli et. al [6] have proposed a personnel selection framework for finding the best possible personnel for a specific job called PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) using a fuzzy logic approach evaluating attributes (experience, foreign language, age, computer knowledge, gender, education, etc) for a specific job and entered into a fuzzy interface of MatLab software, where three type of output is available (rejecting/accepting/pending applicants). Daramola et. al [7] proposed a fuzzy expert system tool for online personnel recruitments, a tool for selection of qualified job applicants with the aim of minimizing the rigor and subjectivity associated with the candidate selection process. Until now main research is based on abilities and talent and not personality.

Fuzzy logic approaches have been important and successful, in software engineering fuzzy based approaches have also been considered like Lather's [9] fuzzy model to evaluate suitability of Software Developers, also Ghasem-Aghaee and Oren's [10] use of fuzzy logic to represent personality for human behavior simulation. Consequently encouraging engineering educators to make greater use of type theory when selecting and forming engineering design teams and delegating team roles, in benefit of achieving productivity and efficiency in team performance.

3 RAMSET Methodology

In our Computer Engineering Program at the University of Baja California in Tijuana Mexico, teaching of Software Engineering is being conducted with development of real software projects applying RAMSET: a Role Assignment Methodology for Software Engineering Teams based on personality, what is unique about our methodology is a combination of Sociometric techniques, Psychometrics and Role Theory in Software Engineering Development Projects, this methodology consists of the next steps: (a) survey for abilities and skills, (b) implementation of personality tests, (c) execute personal interviews, (d) implementation of the sociometric technique, (e) assignment of team roles, (f) follow up of team role fulfillment.

When we developed our RAMSET methodology, we implemented different psychological tests, subjective tests like Myer-Briggs Type Indicator, Big Five and the projective Tree Test. With time and compilation of several cases we have found relationships between personality traits and software engineering roles assigned to people in working teams [11]. RAMSET methodology has been described in previous work documenting information on how to form teams [12] and use of fuzzy approach to find personality patterns, specifically based on Tree Test, Jung and Big Five tests [11][13], thus we are working towards building a Decision Making Fuzzy Model for personnel selection with software support for each test.

This paper specifically analyzes results of the projective Tree Test applied in our singular case studies with RAMSET, not just with arbitrary values but implementing an adaptive neuro-fuzzy inference approach.

4 Tree Test Fuzzy Model

The projective Tree Test used in RAMSET's methodology is personality tests that expresses the relationship between Id, Ego and Super-Ego, and are correlated with drawing attributes root, trunk and crown. Related with part of the root the Id (the It) comprises the personality structure unorganized part that contains basic drives, everything that is inherited and present at birth [15]. Related with the trunk the Ego (the I) constitute the personality structure organized part that includes defensive, perceptual, intellectual-cognitive, and executive functions. Related with the crown is the Super-ego (the Super-I) and aims for perfection, it represents the personality structure organized part mainly but not entirely unconscious, includes ego ideals, spiritual goals and psychic agency (also called 'conscience') that criticizes and prohibits his or her drives, fantasies, feelings and actions.

The perfect equilibrium of these personality instances assures a psychic stability while their disproportion suggests a pathology appearance. The tree's crown represents the subject's fantasies, mental activities, his thoughts, spiritually and reality conception, it covers foliage and branches. The root symbolizes the unconscious world of instincts. Personality's Tree Test throws subjective information, based on the point of view and perception of the evaluator, which is why a Fuzzy Logic Approach has been taken to assess Tree Test uncertainty with numerical values. Fuzzy Inference

Systems are based on Fuzzy Set Theory [16] allowing the incorporation of an uncertainty component that makes them more effective for real approximation. Linguistic variables are used to manipulate imprecise qualitative and quantitative information; the linguistic variable is a variable whose values are not numbers but words or sentences in a natural or artificial language [17]. A linguistic variable is characterized by a quintuple (x, T(x), U, G, M), in which x stands for the name of the variable, T(x) denotes the set of x of fuzzy variable values, ranging over a universe of discourse U. G is a syntactic rule for generating names of x, and M is a semantic rule for associating each x to its meaning being a subset of U.

We selected three input linguistic variables for our Tree FIS; they are (R) Root, (T) Trunk and (F) Foliage; according to the projective Tree Test sketching psycho diagnostic interpretation [18] we can analyze specific drawing characteristics. For Root we can select sketching type and size as it represents the past and reflects person's dependence. For Trunk we can consider form, area, height, sketch intensity and curvature, it depicts the present and reflects person's affectivity. For Foliage we can select form, size and extra features, as it symbolizes achievements or goals reached. We can take into account all these characteristics but some of them are more sensible to define a personality pattern for a person, according to our case studies the most significant characteristics that can identify a personality pattern are sketching of Roots, curvature of Trunk and shape of Foliage. We selected these three characteristics adding type of fruit drawn inside the foliage, although there are more sketch characteristics to consider but if we add them they lower the possibility tro identify a specific personality broadening the range of personalities. The Tree Fuzzy Sets proposed were defined as follows:

The Fuzzy Set of input Linguistic Variable Root is: R(x) = {null, none, with}. When there is no sketch of any root Null is the attribute, if the root is hidden the attribute is None and any sketch of roots the attribute is With.

The Fuzzy Set of input Linguistic Variable Trunk is: T(x) = {straight, wave, trapeze}. When the sketch of the trunk is two parallel lines the attribute is Straight, if one or two of the trunk lines are curved the attribute is Wave, and two straight or curved lines with a wider bottom than the top the attribute is Trapeze.

The Fuzzy Set of input Linguistic Variable Foliage is: F(x) = {circular, cloud, fruit, null}. Just a round sketch of the foliage the attribute is Circular, if it has wavy contour with or without faint sketches inside the attribute is Cloud, if it has any fruits the attribute is Fruit, and any sketch of only branches or leafs the attribute is Null.

The Fuzzy Set of output Linguistic Variable Role is: Q(x)= { Analyst, Architect, Developer-Programmer, Documenter, Tester, Image and Presenter }.

On the first Fuzzy Model we used triangular membership functions as they represent accurately the linguistic terms being modeled, and help parameterization of the model with ease and simplicity, using it as a first fuzzy logic approach to analyze the tree test. Labels were assigned to each attribute of later sets, and consecutive values starting on one were also assigned. For example Root's Set started with a value of 1 assigned to label R1 representing first attribute 'null'; a value of 2 was assigned to

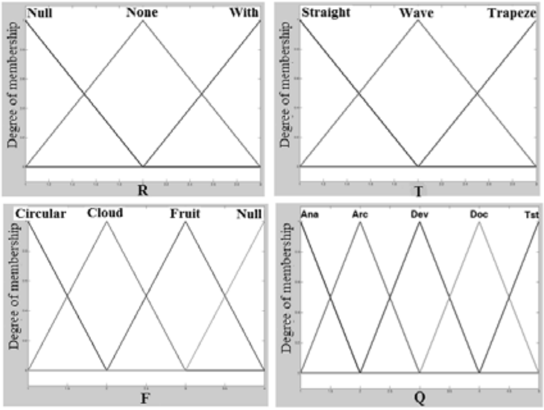


Fig. 1. Membership functions of Tree Test attributes

label R2 representing second attribute 'none'; and a value of 3 was assigned to label R3 that represents last attribute 'with', giving us a universe of discourse from 1 to 3. Figure 1 illustrates attribute's membership functions of linguistic variables Root (R), Trunk (T), Foliage (F) and Role (Q), displaying intervals for each label.

A fuzzy system is associated with a set of rules with meaningful linguistic variables, such as (1)

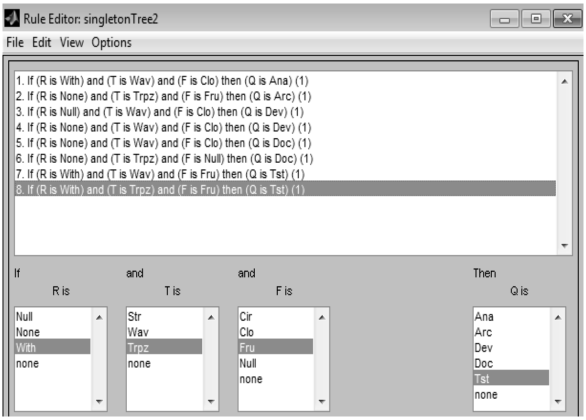
$$R^l \text{ : if } x_1 \text{ is } F_1^{\,l} \text{ and } x_2 \text{ is } F_2^{\,l} \text{ and } \dots\ x_n \text{ is } F_n^{\,l} \quad \text{then } y \text{ is } G^{\,l} \tag{1}$$

Actions are combined with rules in antecedent/consequent format, and then aggregated according to approximate reasoning theory, to produce a nonlinear mapping form input space U = U_1 x U_2 x U_3 x … x U_n to the output space W where $F_k^l \subset U_k$, k = 1,2, …, n are the antecedent membership functions, and $G^{\,l} \subset y$ is the consequent membership function. Input linguistic variables are denoted by u_k , $k = 1,2, \dots, n$, and the output linguistic variable is denoted by y.

The most used FIS models are the Mamdani and Takagi-Sugeno [19]. Mamdani is direct and simple in describing empirical knowledge, has clarity in significance of linguistic variables and design parameters. Takagi-Sugeno enhances a simpler process using first degree equations in most of its applications, at a cost of less clarity in linguistic variables significance. Mamdani fuzzy rules take the form (2), where x and y are activated variables of the membership function, z is the consequent fuzzy variable and the connective AND the conjunction operation with the antecedent.

$$\text{IF x is Xo AND y is Yo THEN z is Zo} \tag{2}$$

With results of our case studies a set of 8 rules were obtained and implemented in MatLab's commercial Fuzzy Logic Toolbox as seen on figure 2.



```
1. If (R is With) and (T is Wav) and (F is Clo) then (Q is Ana) (1)
2. If (R is None) and (T is Trpz) and (F is Fru) then (Q is Arc) (1)
3. If (R is Null) and (T is Wav) and (F is Clo) then (Q is Dev) (1)
4. If (R is None) and (T is Wav) and (F is Clo) then (Q is Dev) (1)
5. If (R is None) and (T is Wav) and (F is Clo) then (Q is Doc) (1)
6. If (R is None) and (T is Trpz) and (F is Null) then (Q is Doc) (1)
7. If (R is With) and (T is Wav) and (F is Fru) then (Q is Tst) (1)
8. If (R is With) and (T is Trpz) and (F is Fru) then (Q is Tst) (1)
```

Fig. 2. Rules of Tree Test Model

5 Tree Test ANFIS Fuzzy Model

Fuzzy Logic Toolbox software computes the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. The Fuzzy Logic Toolbox function that accomplishes this membership function parameter adjustment is called ANFIS. The acronym ANFIS derives its name from Adaptive Neuro-Fuzzy Inference System as defined by Jang [20]. Using a given input/output data set, the toolbox function ANFIS constructs a Fuzzy Inference System (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone or in combination with a least squares type of method. Taking advantage that neuro-adaptive learning techniques provide a method for "learning" information about a data set we also implemented an ANFIS model.

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].

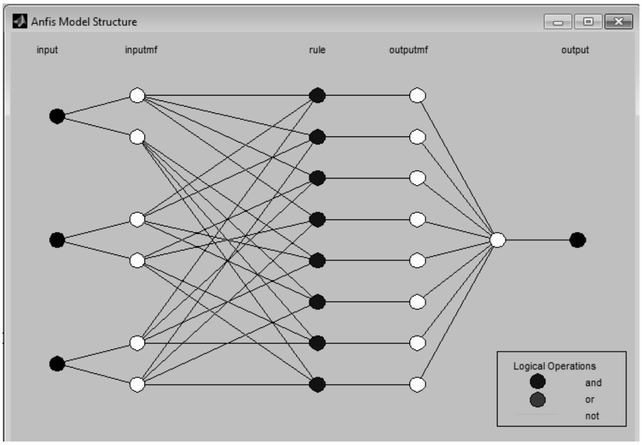


Fig. 3. Tree Test ANFIS Model Architecture with 2 MF's

This paper implemented ANFIS model to the Tree Test, where figure 3 shows our trained ANFIS model architecture using only 2 membership functions. Each sketching characteristic is an input linguistic variable. Root (R) takes a label value of one (1), Trunk (T) a label value of two (2), Foliage (F) a value of three (3). These input variables enter the ANFIS model and obtain an output variable that is the resulting Role recommended; where label values for Role are (1) analyst, (2) architect, (3) developer-programmer, (4) documenter, (5) tester and (6) presenter.

The entire system architecture consists of five layers, these are { input, inputmf, rule, outputmf, output }. Therefore the ANFIS under consideration has three variable inputs denoted by x = { T, R, F }, with two Gaussian membership functions (inputmf denoted by *B*), a set of 8 rules and one output variable Role (R). For a first-order Sugeno Fuzzy Model, a *k*-th rule can be expressed as:

IF (x_1 is B_1^k) AND (x_2 is B_2^k) AND (x_3 is B_3^k) THEN R is $f^k(x)$, where

$$f^k = p_1^k + p_2^k + p_3^k + p_0^k \qquad \forall\ k = 1,2,...,M$$

and membership functions are denoted by:

$$\mu_{B_i^k}(x_i) = \exp\ \left[-\tfrac{1}{2}\left(\frac{x_i - m_i^k}{\sigma_i^k}\right)\right] \tag{3}$$

where p_i^k are linear parameters, and B_i^k are Gaussian membership functions. In our case study architecture (fig. 3) we use 3 input variables (n=3) and 8 rules (M=8), therefore our ANFIS model is defined by:

$$\alpha^k(x_i) = \prod_{i=1}^{n} \mu_{B_i^k}(x_i) \quad \text{; firing strength,}$$

$$\phi^k(x_i) = \frac{\alpha^k}{\sum_{i=1}^{n} \alpha^i}$$; and normalized firing strength, then

$$Q(x_i) = \sum_{k=1}^{M=8} \phi^k(x_i) f^k \tag{4}$$

$$Q(x_i) = \sum_{k=1}^{M=8} \frac{\prod_{i=1}^{n} \mu_{B_i^k}(x_i)}{\sum_{j=1}^{n} \prod_{i=1}^{n} \mu_{B_i^k}(x_i)} \tag{5}$$

where $Q(x_i)$ is the output role in function of vector x = { T, R, F }.

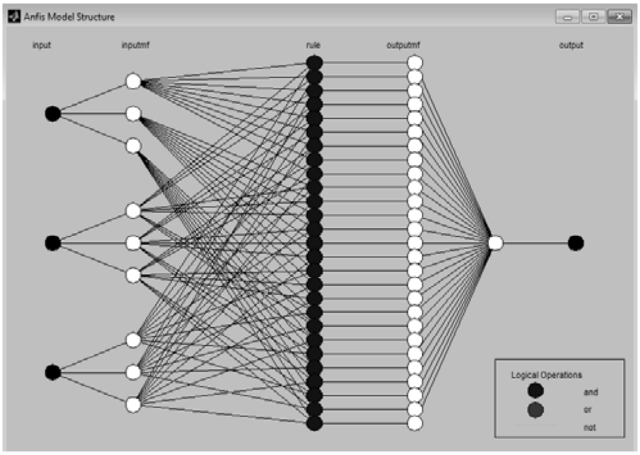


Fig. 4. Tree Test ANFIS Model Architecture with 3 MF's

We also trained an ANFIS model using 3 membership functions and the corresponding equivalent architecture is shown in figure 4, the difference with previous models is a broader integrated quantity measure as this trained ANFIS model obtained 27 rules.

6 Results

Analysis of the Tree Test in a period of 3 years accumulated 74 drawings of trees from software engineering participants. Applying a mean weight method the weights of the attributes of the sketches are presented in table 1 for linguistic variables of Root, Trunk and Foliage respectively. From these fuzzy sets of linguistic variables we can analyze each attribute highlighting for example, when Root (R) is null (R1) the most probable role is Developer-Programmer (Q3). Without a visible root (R2) we can assign Architect (Q2) or Documenter (Q4). Any sketch of root (R3) we are talking about an Analyst (Q1) or Tester (Q5), even Image and Presenter (Q6). The Image and Presenter (Q6) role consists in selling, distribution and image design. The individual's quality performing this role has been related with his own personal image, and a high percentage present the attribute (R3), drawing roots even highlighting thick roots, as we analyze this individual we can see he wants to draw more attention, wants to be noticed and depends of what other people say.

Analyzing the Trunk (T) there are less differences between the roles, wavy trunks (T2) are Analysts (Q1), Developer-Programmers (Q3), Testers (Q5) or Presenters (Q6). What it is sure in this attribute, we can distinguish an Architect (Q2) from the others because he draws the trunk in a trapeze shape (T3). The Foliage (F) distinguishes an Architect (Q2) and a Tester (Q5) from other roles as they draw trees with Fruits (F3), others draw the cloudy (F2) type most of the times.

Table 1. Input Linguistic Variable Weights

Attribute \ Role*	**ANA**	**ARC**	**DEV**	**DOC**	**TST**	**PRS**
R(x)	**ROOT'S WEIGHTS**					
R1	0.103	0.182	0.441	0.030	0.067	0.050
R2	0.276	0.636	0.441	0.727	0.333	0.200
R3	0.621	0.182	0.118	0.242	0.600	0.750
T(x)	**TRUNK'S WEIGHTS**					
T1	0.174	0.174	0.25	0.091	0.097	0.316
T2	0.652	0.174	0.656	0.455	0.452	0.632
T3	0.174	0.652	0.094	0.455	0.452	0.053
F(x)	**FOLIAGE'S WEIGHTS**					
F1	0.225	0.153	0.340	0.243	0.243	0.130
F2	0.6	0.307	0.545	0.540	0.162	0.695
F3	0.15	0.512	0.090	0.162	0.540	0.087
F4	0.025	0.025	0.022	0.054	0.054	0.087
Q(x)*: ANA=Analyst, ARC=Architect, DEV=Developer-programmer, DOC=Documenter, TST=Tester, PRS=Image and Presenter						

Weights from table 1 indicate which attribute is most significant for each Role. With this weights we obtained a Set of Rules were the highest weight is the most significant attribute, therefore the label of that linguistic variable would be the one with the highest value weight. For example an Analyst (label Q1) has label R3 (with), label T2 (wave) and label F2 (cloud) as highest weights, deducing the first rule as:

IF R is R3 AND T is T2 AND F is F2 THEN Q is Q1

Therefore from data of table 1 a Set of Fuzzy Rules is deduced and introduced in our first FIS model as displayed in figure 2. A simple analysis of this set of rules helps us distinguish two roles from others. Architect (Q2) has the only combination of without root (R2), trapeze (T3) and fruits (F3); and the Tester (Q5) is the only one with root (R3), trapeze or wavy (T3 o T2) and fruits (F3). Drawing fruits means this individual has a clear view of what he wants to do, have achieved personal goals in life, giving him the serenity to take charge of any project and achieve goals set and obtain the final product, qualities of a leader and architect.

There's a similarity between developer-programmer (Q3) and documenter (Q4) and between analyst (Q1) and presenter (Q6). Some cases are differentiable between programmer with {R1, T2, F2} and documenter with {R2, T3, F4}, although combination {R2, T2, F2} pops up more frequently. Also the combination {R3, T2, F2} does not distinguish between analyst and presenter, these results give us no significant difference in these cases, thus applying and increasing more cases can give us a more significant result.

Verification of these results is also proven in our different ANFIS models implemented. Comparing our different FIS models, the first ANFIS model with 2 membership functions gives us a short range of output values, only roles 3 and 4 are results for a single input attribute as seen on figure 5.

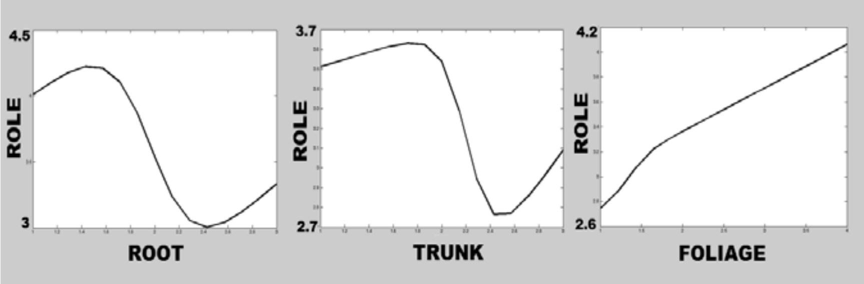


Fig. 5. Input Trait and Output Role Relationships for ANFIS with 2 MF's

Our ANFIS model with 3 membership functions is a better predictor as its range embraces roles 1 thru 4, as seen in figure 6. We corroborate its efficiency when we analyze not just one sketching attribute, buy if we combine and analyze in conjunction. Figure 7 shows the relationship between Root and Trunk, here Role range broadens as every role is considered.

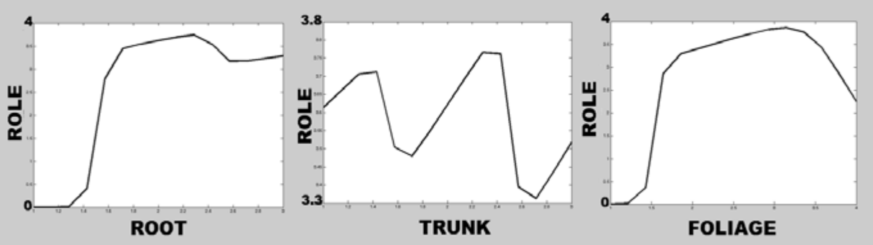


Fig. 6. Input Trait and Output Role Relationships for ANFIS with 3 MF's

The set of rules obtained with ANFIS learning approach implemented in MatLab's commercial Fuzzy Logic Toolbox, help us simulate our case studies and has given us a software support tool to start automating Role Assignment with RAMSET in software engineering projects.

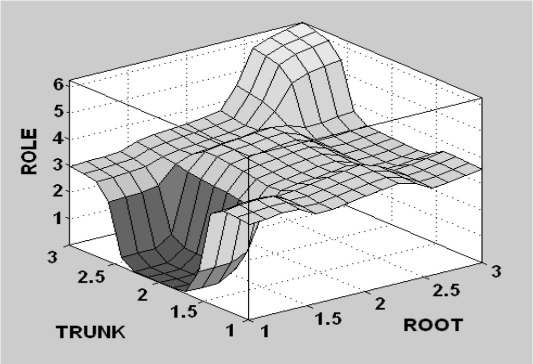


Fig. 7. Root and Trunk Relationships for ANFIS with 3 MF's

7 Conclusions

The objective of using RAMSET is identifying the individual's qualities to perform the most suitable role in the working team. Some personalities and typologies have been identified to perform a type of role; we need more evidence in other type of teams to prove our results applied in software engineering courses established until now. If we work only with the Analyst, Architect and Developer-Programmer roles in our Tree Test Software application, our fuzzy model can help us 100 percent in distinguishing each role. For larger teams that perform with more roles it helps us but we cannot base the role assignment only on the Tree Test, which is why we are proposing the use of other personality tests to complement each other for the best role assignment of the team members.

Implementation of ANFIS models is a highly powerful tool to improve Data Base Rules arisen from this study; combination of different personality test FIS models will create a computer aided software tool invaluable for decision making in assignment of software engineering roles. We know that personality is an important factor to performance of the team, thus is latent the difficulty to assign the adequate role to each member so the team can perform with success.

When working with psychological tests validation is a complex problem because psychology uses statistical tools. Tree Test in psychology is accepted by many psychoanalysts and its validity is cradle in solution of case studies based on interpretation, we are using it in RAMSET to give as a better idea of a person's personality. Problem of role assignment turns out to be so abstract; we are trying to base it on reliable measurements, therefore comparing our results with a reliable test like Big Five, our methodology is being reliable. If we continue with testing and increment of population confidence of our experiment will grow. As we move towards automation of the method interpretation degree is taking out of the equation and a software tool in future development will confirm RAMSET as a methodology for decision making in personnel selection.

References

1. Ahmed, M.A., Muzaffar, Z.: Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic based framework. Information and Software Technology Journal 51(3) (March 2009)
2. Ziv, H., Richardson, D.J.: The Uncertainty Principle in Software Engineering, University of California, Irvine, Technical Report UCI-TR96-33 (August 1996)
3. Bobrow, W.: Personnel Selection and Assessment. The California Psychologist (July/August 2003)
4. Barrick, M.R., Mount, M.K.: The big five personality dimensions and job performance: A meta-analysis. Personnel Psychology 44, 1–26 (1991)
5. Rothstein, M., Goffin, G.R.D.: The use of personality measures in personnel selection: What does current research support? Human Resource Management Review 16(2), 155–180 (2006)
6. 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)
7. Daramola, J.O., Oladipupo, O.O., Musa, A.G.: A fuzzy expert system (FES) tool for online personnel recruitments. Int. J. of Business Inf. Syst. 6(4), 444–462 (2010)
8. Lather, A., Kumar, S., Singh, Y.: Suitability Assessment of Software Developers: A Fuzzy Approach. ACM SIGSOFT Software Engineering Notes 25(3) (May 2000)
9. Oren, T.I., Ghasem-Aghaee, N.: Towards Fuzzy Agents with Dynamic Personality for Human Behavior Simulation. In: SCSC 2003, Montreal PQ, Canada, pp. 3–10 (2003)
10. Martínez, L.G., Rodríguez-Díaz, A., Licea, G., Castro, J.R.: Big Five Patterns for Software Engineering Roles Using An ANFIS Learning Approach with RAMSET. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part II. LNCS, vol. 6438, pp. 428–439. Springer, Heidelberg (2010)
11. Martínez, L.G., Licea, G., Rodríguez-García, A., Castro, J.R.: Experiences in Software Engineering Courses Using Psychometrics with RAMSET. In: ACM SIGCSE ITICSE 2010, Ankara, Turkey, pp. 244–248 (2010)
12. Martínez, L.G., Castro, J.R., Licea, G., Rodríguez-García, A.: Towards a Fuzzy Model for RAMSET: Role Assignment Methodology for Software Engineering Teams. Soft Computing for Intelligent Control and Mobile Robotics 318, 23–41 (2010)
13. Freud, An Outline of Psycho-analysis (1989)
14. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
15. Cox, E.: The Fuzzy Systems Handbook. Academic Press (1994)
16. Koch, K.: El Test del Árbol, Editorial Kapelusz, Buenos Aires (1980)
17. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE TSMC 15, 116–132 (1985)
18. Jang, J.-S.R.: ANFIS: Adaptive Network Based Fuzzy Inference System. IEEE Transactions on Systems, Man, and Cybernetics 23(3) (1993)
19. 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)
20. 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)
21. Hui, H., Song, F.-J., Widjaja, J., Li, J.-H.: ANFIS-based fingerprint matching algorithm. Optical Engineering 43 (2004)
22. Gomathi, V., Ramar, K., Jeeyakumar, A.S.: Human Facial Expression Recognition Using MANFIS Model. Int. J. of Computer Science and Engineering 3(2) (2009)

ACO-Tuning of a Fuzzy Controller for the Ball and Beam Problem

Enrique Naredo and Oscar Castillo

Tijuana Institute of Technology, Tijuana México
ocastillo@tectijuana.mx

Abstract. We describe the use of Ant Colony Optimization (ACO) for the ball and beam control problem, in particular for the problem of tuning a fuzzy controller of the Sugeno type. In our case study the controller has four inputs, each of them with two membership functions; we consider the intersection point for every pair of membership functions as the main parameter and their individual shape as secondary ones in order to achieve the tuning of the fuzzy controller by using an ACO algorithm. Simulation results show that using ACO and coding the problem with just three parameters instead of six, allows us to find an optimal set of membership function parameters for the fuzzy control system with less computational effort needed.

Keywords: Ant Colony Optimization, Fuzzy controller tuning, Fuzzy optimization, ACO optimization for a Fuzzy controller.

1 Introduction

Control systems engineering has an essential role in a wide range of industry processes and over the last few decades the volume of interest in fuzzy controller systems has increased enormously as well as their optimization. Also the development of algorithms for control optimization has been an area of active study, such as Ant Colony Optimization (ACO) which is a bio-inspired population based method, modeling real ant abilities.

This paper proposes to use ACO in order to solve the well known ball and beam benchmark control problem by optimizing a fuzzy logic controller of the Sugeno type. One interesting aspect of this work is the combination of two different techniques from soft computing: Fuzzy Logic as the controller and ACO as the optimizer.

For the fuzzy controller we use the generalized bell function as the membership functions, which have three parameters and because there are two membership functions for every input we need a set of six parameters. Another interesting aspect of this work we use just three parameters instead of six to find an optimal set of membership function parameters with less computational effort needed.

This paper is organized as follows. Section 2 briefly describes related work. Section 3 describes the ball and beam model. In Section 4 the Fuzzy controller is introduced. Problem description is presented in Section 5. Section 6 describes the basic ACO algorithm concepts. Section 7 show experimental results. Finally, conclusions and future studies are presented in the last section.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 58–69, 2011.

2 Related Work

Since fuzzy set theory found an application niche in the control system area, researchers have focus on looking for the optimal set of parameters for a wide range of fuzzy controllers which on the long run will replace the traditional ones.

Optimization can be performed by different kinds of methods, the empirical method is one of the most popular and basically is a methodical approach to trial-and-error basis, there are many others but we are interested on the soft computing based methods. According with Oscar Cordón et al. [15], there are some research works about this issue, such as; pure gradient descent [8][14][18], a mixture of back-propagation and mean least squares estimation, as in ANFIS [11][12], or NEFCLASS (with an NN acting as a simple heuristic) [13], or NN-based on gradient descent method [16], and simulated annealing [1][7][9].

More recent works apply bio-inspired algorithms as optimizers, as in [2][3][17]. The work most related to our paper is [4], on their work they use same control system problem, a fuzzy sliding-mode controller, and we share same type of algorithm as optimizer.

3 Ball and Beam System

The control system used for our purpose is in Fig. 1; the ball and beam system, which is one of the most popular models used for benchmark and research works, this is widely used because of its simplicity.

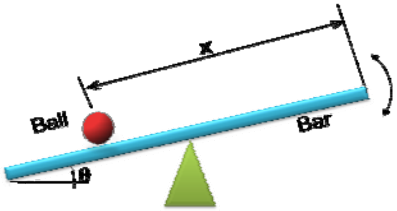


Fig. 1. Ball and beam system

The control task consists on moving the ball to a given position by changing the beam angle, and finally stopping after reaching that position. This system is open loop unstable because the system output (ball position) increases without limit for a fixed input (beam angle) and a feedback control is needed in order to keep the ball in the desired position on the beam.

4 Fuzzy Controller

Because many modern industrial processes are intrinsically unstable, the importance of this type of models becomes relevant in order to test different type of controllers such as the fuzzy ones, Fig.2 shows the block diagram used for the simulation environment software.

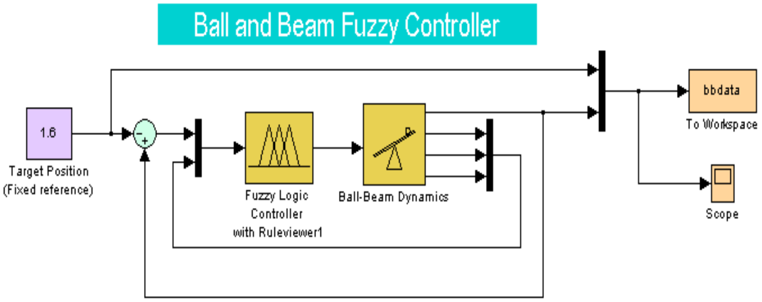


Fig. 2. Model and Fuzzy Controller diagram

A fuzzy controller is a control system based on fuzzy logic, which is widely used in machine control and has the advantage that the solution to the problem can be cast in terms that human operators understand taking advantage of their experience in the controller design.

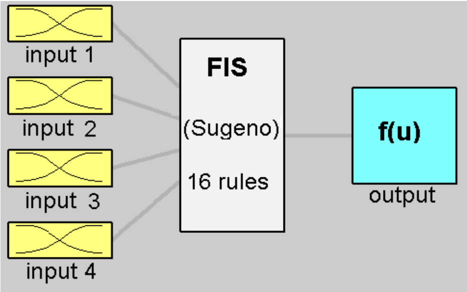


Fig. 3. Fuzzy Inference System

Fig. 3 shows the Fuzzy Inference System (FIS) which has four inputs; ball position x, ball velocity $\dot{x}$, beam angle θ, and beam angle change velocity $\dot{\theta}$, with 16 rules and one output.

5 Problem Description

5.1 Objective

The objective of a tuning process is to adapt a given membership function parameter set, such that the resulting fuzzy controller demonstrates better performance, finding their optimal parameters according with a determined fitness function. Fig 4 shows the architecture of the system used, where ACO is the optimizer used to find a the best set of parameter of the membership functions for the fuzzy controller represented by the FIS, tested into the model in a simulation environment, a cost value is applied using the root mean squared error as the fitness function, and then returned to the algorithm keeping the best so far solutions and trying new paths until the stop criteria is reached.

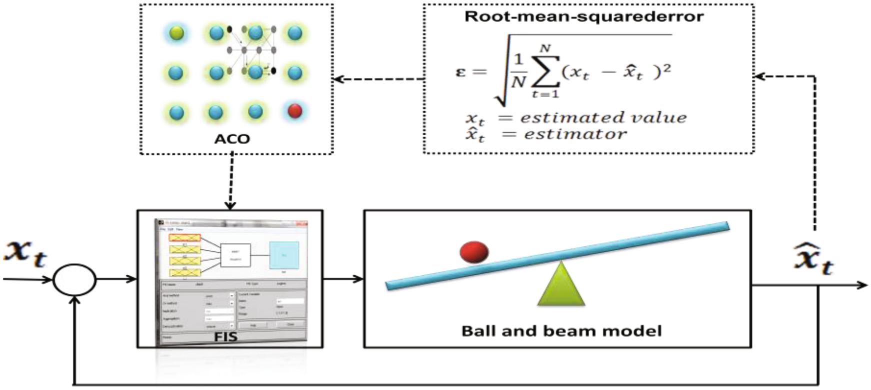


Fig. 4. Architecture of the System

The fitness function establishes the quality of a solution. The measure considered in this case, will be the function called Root Mean Squared Error (RMSE), which is defined in equation 1:

$$\varepsilon = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t - \hat{x}_t\ )^2} \tag{1}$$

where x_t is the estimated value (reference signal), $\hat{x}_t$ is the observed value (control signal), and N is the total observation samples, this is counted not from beginning, starting from the time that the controller shows stable conditions.

5.2 Membership Function

The fuzzy controller has four inputs, each of them has two membership functions and their shape is of generalized bell type, Eq. 2 shows its mathematical notation and Fig. 4(a) shows its graphical representation.

$$f(x, a, b, c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{2}$$

Parameter a represents the standard deviation, b represents the function shape, and c is the center where the function is located, the variation of these three parameters tune the fuzzy controller.

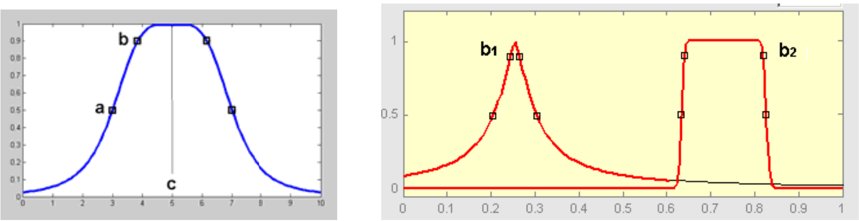


Fig. 5. Generalized bell membership function and its different shapes

Fig. 4(b) shows how the generalized bell membership parameter for b1=0.7 resembles a triangular shape, and for b2=1.5 resembles a square shape.

5.3 Universe of Discourse

For every input we have two membership functions therefore there are six parameter values that define our *universe of discourse.* Because the membership function shape is generally less important than the number of curves and their placement, we consider the intersection point for every pair of membership functions as the main parameter, and their individual shape as secondary ones.

Let define a_1 as the standard deviation of the first membership function for input 1(shown in blue line in Fig. 5), where indexes refer to the function number, and a_2 idem but for the second one (shown in red line in Fig. 5), then we find the intersection point where a_1 on the first membership function right side meets a_2 on the second membership function left side.

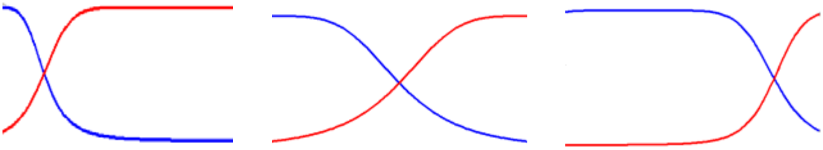


Fig. 6. Membership function intersection point movement

In order to find the intersection point, we let parameters c_1 and c_2 fixed, c_1 take same value for the lower range value, and c_2 same for the upper one. According with this the algorithm chooses a_1 from the set of all possible values given, then compute the a_2 value with the equation 3:

$$a_2 = r - a_1 \tag{3}$$

where r is the range or interval of adjustment, and is given by:

$$r = r_{upper} - r_{lower} \tag{4}$$

The secondary parameters are defined by the individual membership functions shape for every input, given by b_1 and b_2 values. Fig. 6 shows how varying their values we get different shape.

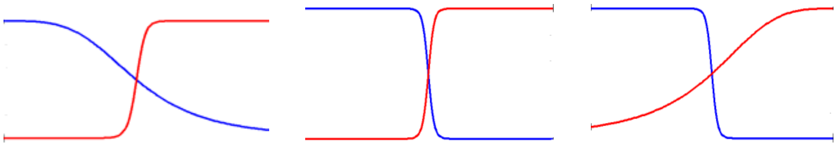


Fig. 7. Membership function shape variation

Getting both main and secondary parameters we have the set of parameters for the membership functions to test into the fuzzy controller.

By coding the problem with just three parameters (one for the intersection point and two for the shape) instead of six, allows us to find for every input an optimal set of membership function parameters for the fuzzy control system with less computational effort needed.

6 Ant Colony Optimization

6.1 ACO Algorithm

In ACO, the information gathered by a single ant is shared among the ant colony and exploited to solve the problem, in this sense ACO acts as a multi-agent approach for solving combinatorial optimization problems, such as the ball and beam problem.

According with Dorigo in [5] and [6], the algorithm shown in Table 1 represents the iterative process of building, evaluating, and updating pheromone that is repeated until a termination condition is met.

In general, the termination condition is either a maximum number of iterations of the algorithm or a stagnation test, which verifies if the solutions created by the algorithm cannot be improved further, an example of this algorithm code can be obtained from [10].

Table 1. ACO algorithm

Pseudocode of a basic ACO algorithm

```
begin
    Initialise();
    while termination condition not met do
        ConstructAntSolution();
        ApplyLocalSearch(); //optional
        UpdatePheromone();
    end
    return bestsolution
end
```

6.2 Heuristic Information

The heuristic information represents a priori information, as we are concern in minimizing the value from the fitness function, and in order to get an heuristic information before running the algorithm, we compute the fitness value from the lower and upper parameters values which represent a vertex or edge on the graph, and then assigning a normalized value to every selected parameter value, obtained by subtracting the lower from the upper and dividing its result by the total number of parameters. Heuristic informationη acts as a short term memory used for ants as relative information from the current node to next node.

6.3 Pheromone

Pheromone is a chemical that ants deposit on the ground when following a certain path while looking for food, this is a form of indirect communication named *stigmergy*, which allows a coordinated behavior in order to find the shortest way from their nest to food. Pheromone τ acts as a long term memory to remember the whole path traversed for every ant.

6.4 Building Solutions

The candidate solutions are created by simulating the movement of artificial ants on the construction graph by moving through neighbor vertices of the construction graph G. The vertices to be visited are chosen in a stochastic decision process, where the probability of choosing a particular neighbor vertex depends on both the problem dependent heuristic information η and the amount of pheromone τassociated with the neighbor vertex (η_j andτ_j, respectively).

An intuitive decision rule is used to select the next vertex to visit, which combines both the heuristic information and the amount of pheromone associated with vertices, this is a decision based on the vertices' probabilities. Given an ant currently located at vertex v_i, the probability of selecting a neighbor vertex v_j is given by

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l\in\mathcal{N}_i^k}[\tau_{il}]^\alpha[\eta_{il}]^\beta}, \ \forall\, j \in \mathcal{N}_i^k \qquad (5)$$

where τ_j and η_j are the pheromone value and heuristic information associated with the j-th vertex, respectively, $\mathcal{N}_i^k$ is the feasible neighborhood of the ant located at vertex v_i (the set of vertices that the ant can visit from v_i), α and β are (user-defined) parameters used to control the influence of the pheromone and heuristic information, respectively.

According to Equation (3), the probability of choosing a particular neighbor vertex is higher for vertices associated with greater amount of pheromone and heuristic information, and subsequently increases in line with increases of the amount pheromone. The pheromone varies as a function of the algorithm iteration t according to how frequent (the more frequent, the higher the pheromone) the vertex or edge has been used in previous candidate solutions.

6.5 Pheromone Trails

After all the ants finished building the candidate solutions of an iteration, the updating of pheromone trails in the construction graph is usually accomplished in two steps, namely reinforcement and evaporation. The reinforcement step consists of increasing the amount of pheromone of every vertex (or edge, in the case that pheromone is associated with edges of the construction graph) used in a candidate solution and it is usually only applied to the best candidate solution.

In general, the pheromone increment is proportional to the quality of the candidate solution, which in turn increases the probability that vertices or edges used in the candidate solution will be used again by different ants. Assuming that pheromone values are associated with vertices of the construction graph, a simple reinforcement rule given by

$$\tau_i = \tau_i + \Delta Q(CS),\ \forall i \in CS \tag{6}$$

where $\Delta\tau_{ij}^{k}$ is the amount of pheromone proportional to the quality of the candidate solution CS to be deposited and τ_i is the pheromone value associated with the i-th vertex of the candidate. For instance, the control optimization is based on the definition of an "odor" associated to each sample represented by an ant and the mutual recognition of ants sharing a similar "odor" to construct a colonial odor used to discriminate between nest mates and intruders. In other approaches, when a specialized ant meets a given object it collects it with a probability that is the higher the sparser are the objects in this region, and after moving, it brings in the object with a probability that is the higher the denser are the objects in this region.

7 Experimental Results

We conducted several experiments using the Ant System algorithm as the optimizer in all cases; Table 2 shows the obtained results. On one hand we use parameter α as the influence weight to select heuristic information; on the other hand we use parameter β as the influence weight to select pheromone trails.

Table 2. ResultAverage Comparison

TYPE	Ants No.	Trails No.	Alpha α	Beta β	Evap. ρ	Iter i	Init.Pher τ	Error ε
AS	10	100	1	2	0.1	100	0.01	0.09877
AS	100	100	1	2	0.1	100	0.01	0.08466
AS	10	1,000	1	2	0.1	100	0.01	0.07430
AS	100	1,000	1	2	0.1	100	0.01	0.07083
AS	10	10,000	1	2	0.1	100	0.01	0.06103
AS	100	10,000	1	2	0.1	100	0.01	0.06083

The number of ants was switched from 10 to 100 every sample, similar criteria was taken for the number of trails from 100 to 1,000.

When running ACO, every iteration of the algorithm choose different membership parameter sets, they were tested, and keeping the best-so-far till reach the maximum number of iterations, obtaining at the end the optimal set for every run. Fig. 8 shows the best error convergence and represents a typical run behavior.

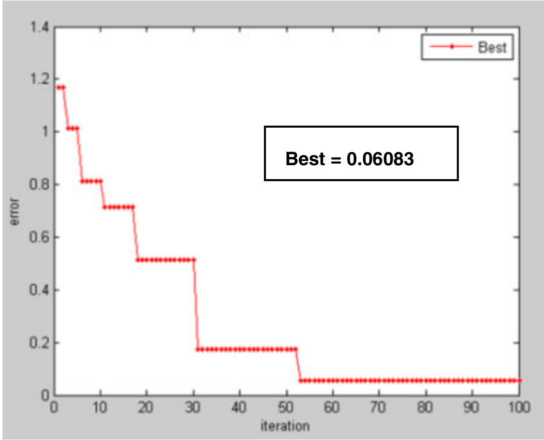


Fig. 8. Best error convergence

The best set of parameters found in our experiments gave us a three parameters for every input for the fuzzy controller and are showed in Fig. 9 where we can note how the intersection point for the input 1 (in1) is displaced to the right hand, while for the input 2 is displaced to the left hand.

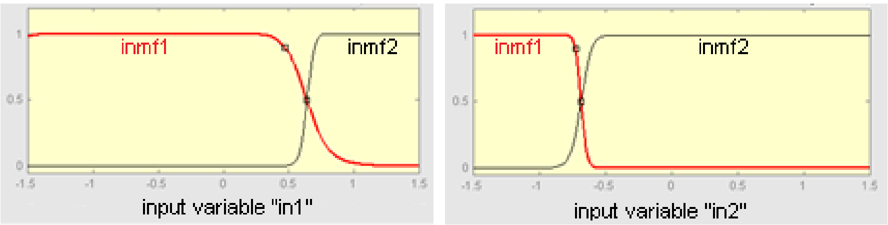


Fig. 9. Best membership functions generated

Inputs 3 and 4 show a little displacement from the middle, as we can see in Fig 10, where was located when using the original controller.

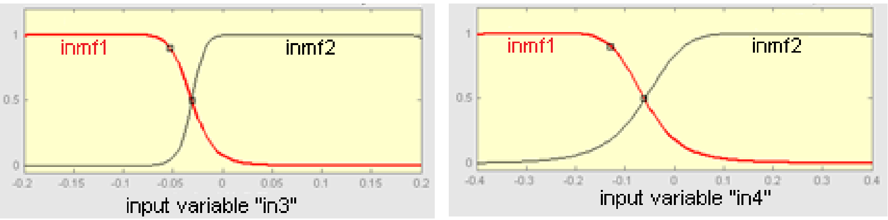


Fig. 10. Best membership functions generated

This shows us graphically how we can find an optimal set of parameters for the fuzzy controller by moving the intersection point as the main parameter and their shape as the secondary parameter for both membership functions.

As the control objective is to reach a desired ball position by moving the beam angle, we observe that the best set of parameters found by ACO meet this objective and Fig. 10 shows the control scope graphic where is the reference is in yellow line and control signal in pink line.

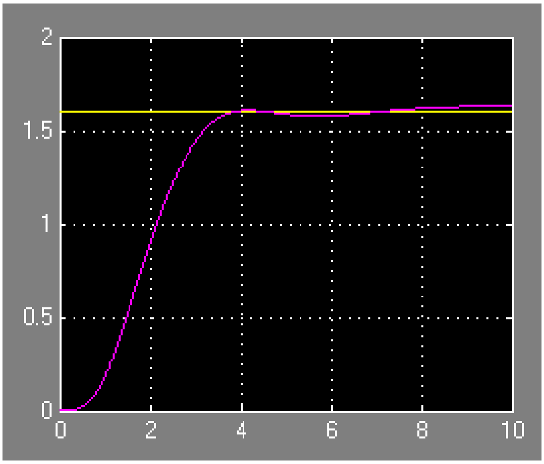


Fig. 11. Control scope graphic

8 Conclusions and Future Studies

We described on this paper how we can use the intersection point for an input pair of membership functions as the main parameter and their individual shape as secondary one to get a simpler representation of the optimization problem.

The fuzzy controller used is of Sugeno type, with four inputs, each of them with two membership functions. The Ant Colony Optimization algorithm was tested to tune the fuzzy controller and simulation results have been shown that ACO works well for the ball and beam control problem.

We can conclude that coding the problem with just three parameters instead of six, and using ACO as the optimizer method allows us to find an optimal set of membership function parameters for the ball and beam fuzzy control system.

For further future work, we propose to try different shapes of membership functions, as well as trying to generalize the intersection point method for more than two membership functions, showing more granularity.

Another direction could be to try in use type-2 fuzzy logic, and adding a particular type of perturbation into the system in order to observe its behavior versus type 1 fuzzy logic. Try different methods, such as; Ant Colony System (ACS), Elitist Ant System (EAS), Ant System Rank (ASrank), MaxMinAnt System (MaxMinAS), Fuzzy Ant Colony Sytem (FACO), etc.

Recent works are concerned in trying to get ACO-hybrid algorithms such as; FACO, PSO-ACO or GA-ACO, it seems to be a good idea trying them on others well known control problems, like bouncing ball, inverted pendulum, flow control, motor control, etc.

Acknowledgment. This work was supported by the National Science and Technology Council from Mexican United States (*Consejo Nacional de Ciencia y Tecnología – CONACYT– de los Estados Unidos Mexicanos*).

References

1. Benitez, J.M., Castro, J.L., Requena, I.: FRUTSA: Fuzzy rule tuning by simulated annealing. To appear in International Journal of Approximate Reasoning (2001)
2. Castillo, O., Martinez-Marroquin, R., Soria, J.: Parameter Tuning of Membership Functions of a Fuzzy Logic Controller for an Autonomous Wheeled Mobile Robot Using Ant Colony Optimization. In: SMC, pp. 4770–4775 (2009)
3. Cervantes, L., Castillo, O.: Design of a Fuzzy System for the Longitudinal Control of an F-14 Airplane. In: Castillo, O., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Intelligent Control and Mobile Robotics. SCI, vol. 318, pp. 213–224. Springer, Heidelberg (2010)
4. Chia-Feng, J., Hao-Jung, H., Chun-Ming, L.: Fuzzy Controller Design by Ant Colony Optimization. IEEE (2007)
5. Dorigo, M., Stützle, T.: Ant Colony Optmization, Massachusetts Institute of Technology. MIT Press (2004)
6. Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.): ANTS 2004. LNCS, vol. 3172. Springer, Heidelberg (2004)
7. Garibaldi, J.M., Ifeator, E.C.: Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation. IEEE Transactions on Fuzzy Systems 7(1), 72–84 (1999)
8. Glorennec, P.Y.: Adaptive fuzzy control. In: Proc. Fourth International Fuzzy Systems Association World Congress (IFSA 1991), Brussels, Belgium, pp. 33–36 (1991)
9. Guely, F., La, R., Siarry, P.: Fuzzy rule base learning through simulated annealing. Fuzzy Sets and Systems 105(3), 353–363 (1999)
10. Haupt, R.L., Haupt, S.E.: Practical Gentic Algorithms, 2nd edn. John Wiley & Sons, Inc. (2004)

11. Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics 23(3), 665–684 (1993)
12. Jang, J.S.R., Sun, C.T., Mizutani, E.: Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall (1997)
13. Nauck, D., Kruse, R.: A neuro-fuzzy method to learn fuzzy classificationrules from data. Fuzzy Sets and Systems 89, 377–388 (1997)
14. Nomura, H., Hayashi, H., Wakami, N.: A self-tuning method of fuzzy control by descendent method. In: Proc. Fourth International Fuzzy Systems Association World Congress (IFSA 1991), Brussels, Belgium, pp. 155–158 (1991)
15. Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic Fuzzy Systems, Evolutionary tuning and learning of fuzzy knowledge bases. In: Advances in Fuzzy Systems-Applications and Theory, pp. 20–25. World Scientific (2000)
16. Shi, Y., Mizumoto, M.: A new approach of neuro-fuzzy learning algorithm for tuning fuzzy rules. Fuzzy Sets and Systems 112, 99–116 (2000)
17. Valdez, F., Melin, P., Castillo, O.: Fuzzy Logic for Parameter Tuning in Evolutionary Computation and Bio-Inspired Methods. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part II. LNCS, vol. 6438, pp. 465–474. Springer, Heidelberg (2010)
18. Vishnupad, P.S., Shin, Y.C.: Adaptive tuning of fuzzy membership functions for nonlinear optimization using gradient descent method. Journal of Intelligent and Fuzzy Systems 7, 13–25 (1999)
19. Yen, J., Langari, R.: Fuzzy Logic: Intelligence, Control and Information, Center for Fuzzy Logic, Robotics, and Intelligent Systems. Texas A&M University, Prentice-Hall (1999)

Estimating Probability of Failure of a Complex System Based on Inexact Information about Subsystems and Components, with Potential Applications to Aircraft Maintenance

Vladik Kreinovich[3], Christelle Jacob[1,2], Didier Dubois[2], Janette Cardoso[1], Martine Ceberio[3], and Ildar Batyrshin[4]

[1] Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), DMIA department,
Campus Supaéro, 10 avenue Édouard Belin, Toulouse, France
jacob@irit.fr, cardoso@isae.fr

[2] Institut de Recherche en Informatique de Toulouse (IRIT), 118 Route de Narbonne
31062 Toulouse Cedex 9, France
dubois@irit.fr

[3] University of Texas at El Paso, Computer Science Dept., El Paso, TX 79968, USA
{mceberio,vladik}@utep.edu

[4] Instituto Mexicano de Petróleo, Ejec Central Lázaro Cardenas Norte 152, Col. San
Bartolo Atepehuacan México D.F., C.P. 07730
batyr@imp.mx

Abstract. In many real-life applications (e.g., in aircraft maintenance), we need to estimate the probability of failure of a complex system (such as an aircraft as a whole or one of its subsystems). Complex systems are usually built with redundancy allowing them to withstand the failure of a small number of components. In this paper, we assume that we know the structure of the system, and, as a result, for each possible set of failed components, we can tell whether this set will lead to a system failure. For each component A, we know the probability $P(A)$ of its failure with some uncertainty: e.g., we know the lower and upper bounds $\underline{P}(A)$ and $\overline{P}(A)$ for this probability. Usually, it is assumed that failures of different components are independent events. Our objective is to use all this information to estimate the probability of failure of the entire the complex system. In this paper, we describe a new efficient method for such estimation based on Cauchy deviates.

Keywords: complex system, probability of failure, interval uncertainty.

1 Formulation of the Problem

It is necessary to estimate the probability of failure for complex systems. In many practical applications, we need to estimate the probability of failure of a complex system. The need for such estimates comes from the fact that in practice, while it is desirable to minimize risk, it is not possible to completely eliminate it: no matter how many precautions we take, there are always some very low probability events that can potentially lead to a system's failure. All we can do is to

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 70–81, 2011.

make sure that the resulting probability of failure does not exceed the desired small value p_0. For example, the probability of a catastrophic event is usually required to be at or below $p_0 = 10^{-9}$.

In aircraft design and maintenance, we need to estimate the probability of a failure of an aircraft as a whole and of its subsystems. At the design stage, the purpose of this estimate is to make sure that this probability of failure does not exceed the allowed probability p_0. At the maintenance stage, this estimate helps to decide whether a maintenance is needed: if the probability of failure exceeds p_0, some maintenance is required to bring this probability down to the desired level p_0 (or below).

Information available for estimating system's probability of failure: general description. Complex systems consist of subsystems, which, in turn, consist of components (or maybe of sub-subsystems which consist of components). So, to estimate the probability of failure of a complex system, we need to take into account when the failure of components and subsystems lead to the failure of the complex system as a whole, and how reliable are these components and subsystems.

From the failure of components and subsystems to the failure of the complex system as a whole. Complex systems are usually built with redundancy allowing them to withstand the failure of a small number of components. Usually, we know the structure of the system, and, as a result, for each possible set of failed components, we can tell whether this set will lead to a system failure. So, in this paper, we will assume that this information is available.

How reliable are components and subsystems? What do we know about the reliability of individual components? For each component A, there is a probability $P(A)$ of its failure. When we have a sufficient statistics of failures of this type of components, we can estimate this probability as the relative frequency of cases when the component failed. Sometimes, we have a large number of such cases, and as a result, the frequency provides a good approximation to the desired probability – so that, in practice, we can safely assume that we know the actual values of these probabilities $P(A)$.

If only a few failure cases are available, it is not possible to get an accurate estimate for $P(A)$. In this case, the only information that we can extract from the observation is the interval $\mathbf{P}(A) = [\underline{P}(A), \overline{P}(A)]$ that contains the actual (unknown) value of this probability.

This situation is rather typical for aircraft design and maintenance, because aircrafts are usually built of highly reliable components – at least the important parts of the aircraft are built of such components – and there are thus very few observed cases of failure of these components.

Component failures are independent events. In many practical situations, failures of different components are caused by different factors. For example, for an aircraft, possible failures of mechanical subsystems can be caused by the material fatigue, while possible failures of electronic systems can be caused by the interference of atmospheric electricity (e.g., when flying close to a thunderstorm).

In this paper, we assume that failures of different components are independent events.

What we do in this paper. Our objective is to use all this information to estimate the probability of failure of the entire complex system. In this paper, we describe a new method for such estimation.

Comment. In this paper, we assumed that failures of different components are independent events. Sometimes, we know that the failures of different components are caused by a common cause; corresponding algorithms are described, e.g., in [1,2,3,8].

2 Simplest Case: Component Failures Are Independent and Failure Probabilities $P(A)$ Are Exactly Known

Let us start our analysis with the simplest case when the component failures are independent and the failure probabilities $P(A)$ for different components A are known exactly. As we mentioned, we assume that there exist efficient algorithms that, given a list of failed components, determines whether the whole system fails or not. In this case, it is always possible to efficiently estimate the probability P of the system's failure by using Monte-Carlo simulations. Specifically, we select the number of simulations N. Then, for each component A, we simulate a Boolean variable *failing(A)* which is true with probability $P(A)$ and false with the remaining probability $1 - P(A)$. This can be done, e.g., if we take the result r of a standard random number generator that generates values uniformly distributed on the interval $[0, 1]$ and select *failing(A)* to be true if $r \le P(A)$ and false otherwise: then the probability of this variable to be true is exactly $P(A)$.

Then, we apply the above-mentioned algorithm to the simulated values of the variables *failing(A)* and conclude whether for this simulation, the system fails or not. As an estimate for the probability of the system's failure, we then take the ratio $p \stackrel{\text{def}}{=} f/N$, where f is the number of simulations on which the system failed. From statistics, it is known that the mean value of this ratio is indeed the desired probability, that the standard deviation can be estimated as $\sigma = \sqrt{p \cdot (1-p)/N} \le 0.5/\sqrt{N}$, and that for sufficiently large N (due to the Central Limit Theorem), the distribution of the difference $P - p$ is close to normal. Thus, with probability 99.9%, the actual value P is within the three-sigma interval $[p - 3\sigma, p + 3\sigma]$.

This enables us to determine how many iterations we need to estimate the probability P with accuracy 10% (and certainty 99.9%): due to $\sigma \le 0.5/\sqrt{N}$, to guarantee that $3\sigma \le 0.1$, it is sufficient to select N for which $3 \cdot 0.5/\sqrt{N} \le 0.1$, i.e., $\sqrt{N} \ge (3 \cdot 0.5)/0.1 = 15$ and $N \ge 225$. It is important to emphasize that this number of iterations is the same no matter how many components we have – and for complex systems, we usually have many thousands of components.

Similarly, to estimate this probability with accuracy 1%, we need $N = 22,500$ iterations, etc. These numbers of iterations work for all possible values P. In practical applications, the desired probability P is small, so $1 - P \approx 1$, $\sigma \approx$

$\sqrt{P/N}$ and the number of iterations, as determined by the condition $3\sigma \le 0.1$ or $3\sigma \le 0.01$, is much smaller: $N \ge 900 \cdot P$ for accuracy 10% and $N \ge 90,000 \cdot P$ for accuracy 1%.

Comment. In many cases, there are also efficient analytical algorithms for computing the desired probability of the system's failure; see, e.g., [4,5,6,16].

3 Important Subcase of the Simplest Case: When Components Are Very Reliable

In many practical applications (e.g., in important subsystems related to aircrafts), components are highly reliable, and their probabilities of failure $P(A)$ are very small. In this case, the above Monte-Carlo technique for computing the probability P of the system's failure requires a large number of simulations, because otherwise, with high probability, in all simulations, all the components will be simulated as working properly.

For example, if the probability of a component's failure is $P(A) = 10^{-3}$, then we need at least a thousand iteration to catch a case when this component fails; if $P(A) = 10^{-6}$, we need at least a million iterations, etc.

In such situations, Monte-Carlo simulations may take a lot of computation time. In some applications, e.g., on the stage of an aircraft design, it may be OK, but in other cases, e.g., on the stage of routine aircraft maintenance, the airlines want fast turnaround, and any speed up is highly welcome.

To speed up such simulations, we can use a re-scaling idea; see, e.g., [8,10]. Specifically, instead of using the original values $P(A)$, we use re-scaled (larger) values $\lambda \cdot P(A)$ for some $\lambda \gg 1$. The value λ is chosen in such a way that the resulting probabilities are larger and thus, require fewer simulations to come up with cases when some components fail. As a result of applying the above Monte-Carlo simulations to these new probabilities $\lambda \cdot P(A)$, we get a probability of failure $P(\lambda)$.

In this case, one can show that while the resulting probabilities $\lambda \cdot P(A)$ are still small, the probability $P(\lambda)$ depends on λ as $P(\lambda) \approx \lambda^k \cdot P$ for some positive integer k.

Thus, to find the desired value P, we repeat this procedure for two different values $\lambda_1 \ne \lambda_2$, get the two values $P(\lambda_1)$ and $P(\lambda_2)$, and then find both unknown k and P from the resulting system of two equations with two unknowns: $P(\lambda_1) \approx \lambda_1^k \cdot P$ and $P(\lambda_2) \approx \lambda_2^k \cdot P$.

To solve this system, we first divide the first equation by the second one, getting an equation $P(\lambda_1)/P(\lambda_2) \approx (\lambda_1/\lambda_2)^k$ with one unknown k, and find $k \approx \ln(P(\lambda_1)/P(\lambda_2))/(\lambda_1/\lambda_2)$. Then, once we know k, we can find P as $P \approx P(\lambda_1)/\lambda_1^k$.

4 Monotonicity Case

Let us start with the simplest subcase when the dependence of the system's failure is monotonic with respect to the failure of components. To be precise, we assume that if for a certain list of failed components, the system fails, it will still fail if we add one more components to the list of failed ones. In this case, the smaller the probability of failure $P(A)$ for each component A, the smaller the probability P that the system as a whole will fail. Similarly, the larger the probability of failure $P(A)$ for each component A, the larger the probability P that the system as a whole will fail.

Thus, to compute the smallest possible value $\underline{P}$ of the failure probability, it is sufficient to consider the values $\underline{P}(A)$. Similarly, to compute the largest possible value $\overline{P}$ of the failure probability, it is sufficient to consider the values $\overline{P}(A)$. Thus, in the monotonic case, to compute the range $[\underline{P}, \overline{P}]$ of possible values of overall failure probability under interval uncertainty, it is sufficient to solve two problems in each of which we know probabilities with certainty:

- to compute $\underline{P}$, we assume that for each component A, the failure probability is equal to $\underline{P}(A)$;
- to compute $\overline{P}$, we assume that for each component A, the failure probability is equal to $\overline{P}(A)$.

5 In Practice, the Dependence Is Sometimes Non-monotonic

In some practically reasonable situations, the dependence of the system's failure on the failure of components is non-monotonic; see, e.g., [8]. This may sound counter-intuitive at first glance: adding one more failing component to the list of failed ones suddenly makes the previously failing system recover, but here is an example when exactly this seemingly counter-intuitive behavior makes perfect sense. Please note that this example is over-simplified: its only purpose is to explain, in intuitive terms, the need to consider non-monotonic case.

To increase reliability, systems include duplication: for many important functions, there is a duplicate subsystem ready to take charge if the main subsystem fails. How do we detect that the main system failed? Usually, a subsystem contains several sensors; sensors sometimes fail, as a result of which their signal no longer reflect the actual value of the quantity they are supposed to measure. For example, a temperature sensor which is supposed to generate a signal proportional to the temperature, if failed, produces no signal at all, which the system will naturally interpret as a 0 temperature. To detect the sensor failure, subsystems often use statistical criteria. For example, for each sensor i, we usually know the mean m_i and the standard deviation σ_i of the corresponding quantity. When these quantities are independent and approximately normally distributed, then, for the measurement values x_i, the sum $X^2 \stackrel{\text{def}}{=} \sum\limits_{i=1}^{n} \dfrac{(x_i - m_i)^2}{\sigma_i^2}$ is the sum of n standard normal distributions and thus, follows the chi-square distributed with n degrees of freedom. So, if the actual value of this sum exceeds the threshold corresponding to confidence level $p = 0.05$, this means that we can confidently conclude that some of the sensors are malfunctioning. If the number n of sensors is large, then one malfunctioning sensor may not increase the sum X^2 too high, and so, its malfunctioning will not be detected, and the system will fail. On the other hand, if all n sensors fail, e.g., show 0 instead of the correct temperature, each term in the sum will be large, the sum will exceed the threshold – and the system will detect the malfunctioning. In this case, the second redundant subsystem will be activated, and the system as a whole will thus continue to function normally.

This is exactly the case of non-monotonicity: when only one sensor fails, the system as a whole fails; however, if, in addition to the originally failed sensor, many other sensors fail, the system as a whole becomes functioning well. Other examples of non-monotonicity can be due to the fact that some components may be in more than two states [9].

In the following text, we will consider the non-monotonic case, in which a simple algorithm (given above) is not applicable.

6 A Practically Important Case When Dependence May Be Non-monotonic but Intervals Are Narrow: Towards a New Algorithm

General non-monotonic case: a possible algorithm. For each component A, by using the formula of full probability, we can represent the probability P of the system's failure as follows:

$$P = P(A) \cdot P(F|A) + (1 - P(A)) \cdot P(F|\neg A),$$

where $P(F|A)$ is the conditional probability that the system fails under the condition that the component A fails, and $P(F|\neg A)$ is the conditional probability that the system fails under the condition that the component A does not fail. The conditional probabilities $P(F|A)$ and $P(F|\neg A)$ do not depend on $P(A)$, so the resulting dependence of P on $P(A)$ is linear. A linear function attains it minimum and maximum at the endpoints. Thus, to find $\underline{P}$ and $\overline{P}$, it is not necessary to consider all possible values $P(A) \in [\underline{P}(A), \overline{P}(A)]$, it is sufficient to only consider two values: $P(A) = \underline{P}(A)$ and $P(A) = \overline{P}(A)$.

For each of these two values, for another component A', we have two possible options $P(A') = \underline{P}(A')$ and $P(A') = \overline{P}(A')$; thus, in this case, we need to consider $2 \times 2 = 4$ possible combinations of values $P(A)$ and $P(A')$.

In general, when we have k components $A_1, \ldots, A_k$, it is sufficient to consider 2^k possible combinations of values $\underline{P}(A_i)$ and $\overline{P}(A_i)$ corresponding to each of these components. This procedure requires times which grows as 2^k. As we mentioned earlier, when k is large, the needed computation time becomes unrealistically large.

Natural question. The fact that the above algorithm requires unrealistic exponential time raises a natural question: is it because our algorithm is inefficient or is it because the problem itself is difficult?

The problem is NP-hard. In the general case, when no assumption is made about monotonicity, the problem is as follows:

– Let F be a propositional formula with n variables A_i
– for each variable A_i, we know the interval $[\underline{P}(A_i), \overline{P}(A_i)]$ that contains the actual (unknown) $P(A_i)$ that this variable is true;
– we assume that the Boolean variables are independent.

Different values $P(A_i) \in [\underline{P}(A_i), \overline{P}(A_i)]$ lead, in general, to different values of the probability P that F is true (e.g., that the system fails). Our objective is to compute the range $[\underline{P}, \overline{P}]$ of possible values of this probability.

In [8], we have proven that, in general, the problem of computing the desired range $[\underline{P}, \overline{P}]$ is NP-hard. From the practical viewpoint, this means, that (unless P=NP, which most computer scientists believe to be not true), there is no hope to avoid non-feasible exponential time. Since we cannot have a feasible algorithm that is applicable to all possible cases of the general problem, we therefore need to restrict ourselves to practically important cases – and try to design efficient algorithms that work for these cases. This is what we do in this paper.

A practically important case of narrow intervals. When there is enough information, the intervals $[\underline{P}(A), \overline{P}(A)]$ are narrow. If we represent them in the form

$$[\widetilde{P}(A) - \Delta(A), \widetilde{P}(A) + \Delta(A)],$$

with $\widetilde{P}(A) = \dfrac{\underline{P}(A) + \overline{P}(A)}{2}$ and $\Delta(A) = \dfrac{\overline{P}(A) - \underline{P}(A)}{2}$, then values $\Delta(A)$ are small, so we can safely ignore terms which are quadratic or of higher order in terms of $\Delta P(A)$.

Linearization: analysis of the problem. In the case of narrow intervals, the difference $\Delta P(A) \stackrel{\text{def}}{=} P(A) - \widetilde{P}(A)$ is bounded by $\Delta(A)$ and thus, also small: $|\Delta P(A)| \le \Delta(A)$. Hence, we can expand the dependence of the desired system failure probability $P = P(P(A), \ldots) = P(\widetilde{P}(A) + \Delta P(A), \ldots)$ into Taylor series and keep only terms which are linear in $\Delta P(A)$: $P \approx \widetilde{P} + \sum\limits_A c_A \cdot \Delta P(A)$, where $\widetilde{P} \stackrel{\text{def}}{=} P(\widetilde{P}(A), \ldots)$ and $c_A \stackrel{\text{def}}{=} \dfrac{\partial}{\partial P(A)} P(\widetilde{P}(A), \ldots)$.

For those A for which $c_A \ge 0$, the largest value of the sum $\sum\limits_A c_A \cdot \Delta P(A)$ (when $\Delta P(A) \in [-\Delta(A), \Delta(A)]$) is attained when $\Delta P(A)$ attains its largest possible value $\Delta(A)$. Similarly, when $c_A < 0$, the largest possible values of the sum is attained when $\Delta P(A) = -\Delta(A)$. In both cases, the largest possible value of the term $c_A \cdot \Delta P(A)$ is $|c_A| \cdot \Delta(A)$. Thus, the largest possible value of P is equal to $\widetilde{P} + \Delta$, where

$$\Delta \stackrel{\text{def}}{=} \sum_A |c_A| \cdot \Delta(A).$$

Similarly, one can show that the smallest possible value of P is equal to $\widetilde{P} - \Delta$, so the range of possible values of the failure probability P is $[\widetilde{P} - \Delta, \widetilde{P} + \Delta]$.

We already know how to compute $\widetilde{P}$ – e.g., we can use the Monte-Carlo approach. How can we compute Δ?

How to compute Δ: numerical differentiation and its limitations. A natural idea is to compute all the partial derivatives c_A and to use the above formula for Δ. By definition, c_A is the derivative, i.e.,

$$c_A = \lim_{h \to 0} \frac{P(\widetilde{P}(A) + h, \widetilde{P}(B), \widetilde{P}(C), \ldots) - P(\widetilde{P}(A), \widetilde{P}(B), \widetilde{P}(C), \ldots)}{h}.$$

By definition of the limit, this means that to get a good approximation for c_A, we can take a small h and compute

$$c_A = \frac{P(\widetilde{P}(A) + h, \widetilde{P}(B), \widetilde{P}(C), \ldots) - P(\widetilde{P}(A), \widetilde{P}(B), \widetilde{P}(C), \ldots)}{h}.$$

This approach to computing derivatives is called *numerical differentiation.*

The problem with this approach is that each computation of the value $P(\widetilde{P}(A) + h, \widetilde{P}(B), \widetilde{P}(C), \ldots)$ by Monte-Carlo techniques requires a lot of simulations, and we need to repeat these simulations again and again as many times as there are components. For an aircraft, with thousands of components, the resulting increase in computation time is huge. Moreover, since we are interested in the difference $P(\widetilde{P}(A) + h, \ldots) - P(\widetilde{P}(A), \ldots)$ between the two probabilities, we need to compute each of these probabilities with a high accuracy, so that this difference would be visible in comparison with the approximation error $\sim 1/\sqrt{N}$ of the Monte-Carlo estimates. This requires that we further increase the number of iterations N in each Monte-Carlo simulation and thus, even further increase the computation time.

Cauchy deviate techniques: reminder. In order to compute the value $\sum\limits_A |c_A| \cdot \Delta(A)$ faster, one may use a technique based on *Cauchy distributions* (e.g., [12,15]) , i.e., probability distributions with probability density of the form $\rho(z) = \dfrac{\Delta}{\pi \cdot (z^2 + \Delta^2)}$; the value Δ is called the *scale parameter* of this distribution, or simply a *parameter*, for short.

Cauchy distribution has the following property: if z_A corresponding to different A are independent random variables, and each z_A is distributed according to the Cauchy law with parameter $\Delta(A)$, then their linear combination $z = \sum\limits_A c_A \cdot z_A$ is also distributed according to a Cauchy law, with a scale parameter $\Delta = \sum\limits_A |c_A| \cdot \Delta(A)$.

Therefore, using Cauchy distributed random variables δ_A with parameters $\Delta(A)$, the difference

$$c \stackrel{\text{def}}{=} P(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots) - P(\widetilde{P}(A), \widetilde{P}(B), \ldots) = \sum_A c_A \cdot \delta_A$$

is Cauchy distributed with the desired parameter Δ. So, repeating this experiment N_c times, we get N_c values $c^{(1)}, \ldots, c^{(N_c)}$ which are Cauchy distributed with the unknown parameter, and from them we can estimate Δ. The bigger N_c, the better estimates we get.

Comment. To avoid confusion, we should emphasize that the use of Cauchy distributions is a computational technique, *not* an assumption about the actual distribution: indeed, we know that the actual value of $\Delta P(A)$ is bounded by $\Delta(A)$, but for a Cauchy distribution, there is a positive probability that the simulated value is larger than $\Delta(A)$.

Cauchy techniques: towards implementation. In order to implement the above idea, we need to answer the following two questions:

– how to simulate the Cauchy distribution;
– how to estimate the parameter Δ of this distribution from a finite sample.

Simulation can be based on the functional transformation of uniformly distributed sample values: $\delta_A = \Delta(A) \cdot \tan(\pi \cdot (r_A - 0.5))$, where r_A is uniformly distributed on the interval $[0, 1]$.

In order to estimate Δ, we can apply the Maximum Likelihood Method

$$\rho(c^{(1)}) \cdot \rho(c^{(2)}) \cdot \ldots \cdot \rho(c^{(N_c)}) \to \max,$$

where $\rho(z)$ is a Cauchy distribution density with the unknown Δ. When we substitute the above-given formula for $\rho(z)$ and equate the derivative of the product with respect to Δ to 0 (since it is a maximum), we get an equation

$$\frac{1}{1+\left(\frac{c^{(1)}}{\Delta}\right)^2} + \ldots + \frac{1}{1+\left(\frac{c^{(N_c)}}{\Delta}\right)^2} = \frac{N_c}{2}.$$

Its left-hand side is an increasing function that is equal to $0 (< N_c/2)$ for $\Delta = 0$ and $> N_c/2$ for $\Delta = \max \left|c^{(k)}\right|$; therefore the solution to this equation can be found by applying a bisection method to the interval $\left[0, \max \left|c^{(k)}\right|\right]$.

It is important to mention that we assumed that the function P is reasonably linear when the values δ_A are small: $|\delta_A| \le \Delta(A)$. However, the simulated values δ_A may be larger than $\Delta(A)$. When we get such values, we do not use the original function P for them, we use a normalized function that is equal to P within the given intervals, and that is extended linearly for all other values; we will see, in the description of an algorithm, how this is done.

Cauchy deviate technique: main algorithm

– Apply P to the values $\widetilde{P}(A)$ and compute $\widetilde{P} = P(\widetilde{P}(A), \widetilde{P}(B), \ldots)$.
– For $k = 1, 2, \ldots, N_c$, repeat the following:
    - use the standard random number generator to compute n numbers $r_A^{(k)}$ that are uniformly distributed on the interval $[0, 1]$;
    - compute Cauchy distributed values $c_A^{(k)} = \tan(\pi \cdot (r_A^{(k)} - 0.5))$;
    - compute the largest value of $|c_A^{(k)}|$ so that we will be able to normalize the simulated measurement errors and apply P to the values that are within the box of possible values: $K = \max\limits_A |c_A^{(k)}|$;
    - compute the simulated measurement errors $\delta_A^{(k)} := \Delta(A) \cdot c_A^{(k)}/K$;

  - compute the simulated probabilities $P^{(k)}(A) = \widetilde{P}(A) + \delta_A^{(k)}$;
  - estimate $P(P^{(k)}(A), P^{(k)}(B), \ldots)$ and then compute

$$c^{(k)} = K \cdot (P(P^{(k)}(A), P^{(k)}(B), \ldots) - \widetilde{P});$$

- Compute $\varDelta$ by applying the bisection method to solve the corresponding equation.

Resulting gain and remaining limitation. By using the Monte-Carlo techniques, we make sure that the number of iterations N_c depends only on the accuracy with which we want to find the result and not on the number of components. Thus, when we have a large number of components, this method is faster than numerical differentiation.

The computation time of the new algorithm is smaller, but it is still not very fast. The reason is that the Cauchy method was originally was designed for situations in which we can compute the exact value of $P(P^{(k)}(A), P^{(k)}(B), \ldots)$. In our problem, these values have to be computed by using Monte-Carlo techniques, and computed accurately – and each such computation requires a lot of iterations. Instead of running the maximum likelihood, we can also just estimate $\varDelta$ by means of the sample interquartile range instead of solving the non-linear equation. But this method will be less accurate.

Final idea to further decrease the needed number of simulations. (see, e.g., Section 5.4 of [15]) For each combination of values δ_A, the corresponding Monte-Carlo simulation produces not the actual probability $P(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots)$, but an *approximate* value $\widetilde{P}(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots) = P(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots) + c_n$ that differs from the desired probability by a random variable c_n which is normally distributed with mean 0 and variance $\sigma^2 = \dfrac{\widetilde{P} \cdot (1 - \widetilde{P})}{N}$. As a result, the difference $c \stackrel{\text{def}}{=} \widetilde{P}(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots) - \widetilde{P}$ between the two observed probabilities can be represented as $c = c_c + c_n$, where $c_c \stackrel{\text{def}}{=} P(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots) - \widetilde{P}$ is, as we have mentioned, Cauchy distributed with parameter $\varDelta$, while

$$c_n = \widetilde{P}(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots) - P(\widetilde{P}(A) + \delta_A, \widetilde{P}(B) + \delta_B, \ldots)$$

is normally distributed with mean 0 and known standard deviation σ.

The components c_c and c_n are independent. Thus, for $c = c_c + c_n$, for the characteristic function $\chi(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot c)]$, we have

$$E[\exp(\mathrm{i} \cdot \omega \cdot c)] = E[\exp(\mathrm{i} \cdot \omega \cdot c_c) \cdot \exp(\mathrm{i} \cdot \omega \cdot c_n)] = E[\exp(\mathrm{i} \cdot \omega \cdot c_c)] \cdot E[\exp(\mathrm{i} \cdot \omega \cdot c_n)],$$

i.e., $\chi(\omega) = \chi_c(\omega) \cdot \chi_n(\omega)$, where $\chi_c(\omega)$ and $\chi_n(\omega)$ are characteristic functions of c_c and c_n. For Cauchy distribution and for the normal distribution, the characteristic functions are known: $\chi_c(\omega) = \exp(-|\omega| \cdot \varDelta)$ and $\chi_n(\omega) = \exp(-\omega^2 \cdot \sigma^2)$. So, we conclude that $\chi(\omega) = \exp(-|\omega| \cdot \varDelta - \omega^2 \cdot \sigma^2)$. Hence, to determine $\varDelta$, we can estimate $\chi(\omega)$, compute its negative logarithm, and then compute $\varDelta$ (see the formula below).

Since the value $\chi(\omega)$ is real, it is sufficient to consider only the real part $\cos(\ldots)$ of the complex exponent $\exp(\mathrm{i}\cdot\ldots)$. Thus, we arrive at the following algorithm:

Algorithm. First, we use a lengthy Monte-Carlo simulation to compute the value. $\widetilde{P} = P(\widetilde{P}(A), \widetilde{P}(B), \ldots)$. Then, for $k = 1, 2, \ldots, N$, we repeat the following:

– use a random number generator to compute n numbers $r_A^{(k)}$, that are uniformly distributed on the interval $[0, 1]$;
– compute $\delta_A^{(k)} = \Delta_i \cdot \tan(\pi \cdot (r_A^{(k)} - 0.5))$;
– use Monte-Carlo simulations to find the frequency (probability estimate) $\widetilde{P}(\widetilde{P}(A) + \delta_A^{(k)}, \widetilde{P}(B) + \delta_B^{(k)}, \ldots)$ and then

$$c^{(k)} = \widetilde{P}(\widetilde{P}(A) + \delta_A^{(k)}, \widetilde{P}(B) + \delta_B^{(k)}, \ldots) - \widetilde{P};$$

– for a real number $\omega > 0$, compute $\chi(\omega) = \dfrac{1}{N} \cdot \sum\limits_{k=1}^{N} \cos\left(\omega \cdot c^{(k)}\right)$;
– compute $\Delta = -\dfrac{\ln(\chi(\omega))}{\omega} - \sigma^2 \cdot \dfrac{\omega}{2}$.

Comment. Of course, we also need, as before, to "reduce" the simulated values δ_A to the given bounds $\Delta(A)$.

7 Conclusion

In this paper we considered the problem of estimating the probability of failure P of a complex system such as an aircraft, assuming we only know upper and lower bounds of probabilities of elementary events such as component failures. The assumptions in this paper is that failures of different components are independent events, and that there is enough information to ensure narrow probability intervals. The problem of finding the resulting range $[\underline{P}, \overline{P}]$ of possible values of P is computationally difficult (NP-hard). In this paper, for the practically important case of narrow intervals $[\underline{P}(A), \overline{P}(A)]$, we propose an efficient method that uses Cauchy deviates to estimate the desired range $[\underline{P}, \overline{P}]$. Future works concern the estimation of intervals $[\underline{P}(A), \overline{P}(A)]$ from the imprecise knowledge of failure rates. Moreover, it is interesting to study what can be done in practice when the independence assumption on component failures no longer holds.

Acknowledgments. C. Jacob was supported by a grant from @MOST Prototype, a joint project of Airbus, LAAS-CNRS, ONERA, and ISAE. V. Kreinovich was supported by the Nat'l Science Foundation grants HRD-0734825 and DUE-0926721 and by Grant 1 T36 GM078000-01 from the Nat'l Institutes of Health. We are thankful to the anonymous referees for valuable suggestions.

References

1. Ceberio, M., et al.: Interval-type and affine arithmetic-type techniques for handling uncertainty in expert systems. Journal of Computational and Applied Mathematics 199(2), 403–410 (2007)
2. Chopra, S.: Affine arithmetic-type techniques for handling uncertainty in expert systems, Master's thesis, Department of Computer Science, University of Texas at El Paso (2005)
3. Chopra, S.: Affine arithmetic-type techniques for handling uncertainty in expert systems. International Journal of Intelligent Technologies and Applied Statistics 1(1), 59–110 (2008)
4. Dutuit, Y., Rauzy, A.: Approximate estimation of system reliability via fault trees. Reliability Engineering and System Safety 87(2), 163–172 (2005)
5. Flage, R., et al.: Handling of epistemic uncertainties in fault tree analysis: a comparison between probabilistic, possibilistic, and hybrid approaches. In: Bris, S., Guedes Sares, C., Martorell, S. (eds.) Proc. European Safety and Reliability Conf. Reliability, Risk and Safety: Theory and Applications, ESREL 2009, Prague, September 7-10, 2009 (2010)
6. Guth, M.A.: A probability foundation for vagueness and imprecision in fault tree analysis. IEEE Transations on Reliability 40(5), 563–570 (1991)
7. Interval computations website, http://www.cs.utep.edu/interval-comp
8. Jacob, C., et al.: Estimating probability of failure of a complex system based on partial information about subsystems and components, with potential applications to aircraft maintenance. In: Proc. Int'l Workshop on Soft Computing Applications and Knowledge Discovery SCAKD 2011, Moscow, Russia, June 25 (2011)
9. Jacob, C., Dubois, D., Cardoso, J.: Uncertainty Handling in Quantitative BDD-Based Fault-Tree Analysis by Interval Computation. In: Benferhat, S., Grant, J. (eds.) SUM 2011. LNCS, vol. 6929, pp. 205–218. Springer, Heidelberg (2011)
10. Jaksurat, P., et al.: Probabilistic approach to trust: ideas, algorithms, and simulations. In: Proceedings of the Fifth International Conference on Intelligent Technologies InTech 2004, Houston, Texas, December 2-4 (2004)
11. Jaulin, L., et al.: Applied Interval Analysis. Springer, London (2001)
12. Kreinovich, V., Ferson, S.: A new Cauchy-based black-box technique for uncertainty in risk analysis. Reliability Engineering and Systems Safety 85(1-3), 267–279 (2004)
13. Kreinovich, V., et al.: Computational Complexity and Feasibility of Data Processing and Interval Computations. Kluwer, Dordrecht (1997)
14. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM Press, Philadelphia (2009)
15. Trejo, R., Kreinovich, V.: Error estimations for indirect measurements: randomized vs. deterministic algorithms for 'black-box' programs. In: Rajasekaran, S., et al. (eds.) Handbook on Randomized Computing, pp. 673–729. Kluwer (2001)
16. Troffaes, M., Coolen, F.: On the use of the imprecise Dirichlet model with fault trees. In: Proceedings of the Mathematical Methods in Reliability Conference, Glasgow (July 2007)
17. Walley, P.: Statistical reasoning with imprecise probabilities. Chapman & Hall, New York (1991)

Two Steps Individuals Travel Behavior Modeling through Fuzzy Cognitive Maps Pre-definition and Learning

Maikel León[1,2], Gonzalo Nápoles[1], María M. García[1],
Rafael Bello[1], and Koen Vanhoof[2]

[1] Central University of Las Villas, Santa Clara, Cuba
mle@uclv.edu.cu
[2] Hasselt University, Diepenbeek, Belgium

Abstract. Transport "management and behavior" modeling takes place in developed societies because of the benefit that it brings for all social and economic processes. Using in this field, advanced computer science techniques like Artificial Intelligence is really relevant from the scientific, economic and social point of view. This paper deals with Fuzzy Cognitive Maps as an approach in representing the behavior and operation of such complex systems. Two steps are presented, an initial modeling trough Automatic Knowledge "Engineering and Formalizing"; and secondly, using readjustment of parameters with an inspired on Particle Swarm Optimization learning method. The theoretical results come from necessities in a real case study that is also presented, showing then the practical approach of the proposal, where new issues were obtained but also real problems were solved.

Keywords: Fuzzy Cognitive Maps, Particle Swarm Optimization, Simulation, Travel Behavior, Decision Making.

1 Introduction

Transport Demand Management (TDM) is of vital importance for decreasing travel-related energy consumption and depressing high weight on urban infrastructure. Also known as "mobility management", is a term for measures or strategies to make improved use of transportation means by reducing travel demand or distributing it in time and space. Many attempts have been made to enforce TDM that would influence individuals unsustainable travel behavior towards more sustainable forms, however TDM can be effectively and efficiently implemented if they are developed founded on a profound understanding of the basic causes of travel, such as people's reasons and inclinations, and comprehensive information of individuals behaviors [1].

In the process of transportation planning, TDM forecast is one of the most important analysis instruments to evaluate various policy measures aiming at influencing travel supply and demand. In past decades, increasing environmental awareness and the generally accepted policy paradigm of sustainable development made transportation policy measures shift from facilitation to reduction and control. Objectives of TDM

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 82–94, 2011.

measures are to alter travel behavior without necessarily embarking on large-scale infrastructure expansion projects, to encourage better use of available transport resources avoiding the negative consequences of continued unrestrained growth in private mobility. Individual activity travel choices can be considered as actual decision problems, causing the generation of a mental representation or cognitive map of the decision situation and alternative courses of action in the expert's mind. This cognitive map concept is often referred to in theoretical frameworks of travel demand models, especially related to the representation of spatial dimensions [2].

However, actual model applications are scarce, mainly due to problems in measuring the construct and putting it into the model's operation. The development of the mental map concept can benefit from the knowledge provided by individual tracking technologies. Researches are focusing on that direction, in order to improve developed models and to produce a better quality of systems. At an individual level it is important to realize that the relationship between travel decisions and the spatial characteristics of the environment is established through the individual's perception and cognition of space. As an individual observes space, for instance through travel, the information is added to the individual's mental maps [3].

Records regarding individual's decision making processes can be used as input to generate mental models. Such models treat each individual as an agent with mental qualities, such as viewpoints, objectives, predilections, inclinations, etc. For the modeling of such models, several Artificial Intelligence (AI) techniques can be used; in this case Fuzzy Cognitive Maps (FCM) will be study. These maps try to genuinely simulate individual's decision making processes. Consequently, can be used not only to understand people's travel behaviors, but also to pretend the changes in their actions, due to some factors in their decision atmosphere. This technique is very well known by its "self-explicability".

More computationally speaking, FCM are a combination of Fuzzy Logic and Neural Networks; combining the heuristic and common sense rules of Fuzzy Logic with the learning heuristics of the Neural Networks. They were introduced by the famous scientific B. Kosko, who enhanced cognitive maps with fuzzy reasoning, that had been previously used in the field of socio-economic and political sciences to analyze social decision-making problems, etc. The use of FCM for many applications in different scientific fields was proposed. FCM had been apply to analyze extended graph theoretic behavior, to make decision analysis and cooperate distributed agents, were used as structures for automating human problem solving skills and as behavioral models of virtual worlds, etc.

In this present work, FCM constitute a good alternative to study individuals during their decision making process. A decision maker activates a temporary mental representation in his/her working memory based on his/her previous experiences or existing knowledge. Therefore, constructing a mental representation requires a decision maker to recall, reorder and summarize relevant information in his memory. It may involve translating and representing this information into other forms, such as a scheme or diagram, supporting coherent reasoning in a connected structure.

More specific, our case investigation takes place in the city of Hasselt, capital of the Flemish province of Limburg, Belgium, a study related to Travel Behavior has been made. The city has a population around 72 000 habitants, with a traffic junction of important traffic arteries from all directions. All the city's local zero-fare buses, Hasselt made public transport by bus zero-fare from 1 July 1997 and buses use was said to be as much as "15 times higher" by 2010, being the first city in the world that had entirely zero-fare bus services on the whole of its territory.

This paper will present our proposed approach for the generating of FCM as a knowledge representation form for the modeling of individuals decision making mental structures concerning travel activities. Once the problem is presented, the data gathering process is described, and the steps for the construction of the model will be explained. Application software will be also offered and at the end, validation and reflection sections conclude the contribution.

2 Data Gathering Process through Knowledge Engineering

Knowledge Engineering (KE) is defined as the group of principles, methods and tools that allow applying the scientific knowledge and experience by means of useful constructions for the human. It faces the problem of building computational systems with dexterity, aspiring first to acquire the knowledge of different sources and, in particular, to conclude the knowledge of the expert ones and then to organize them in an effective implementation. The KE is the process to design and make operative the Knowledge Based Systems (KBS); it is the topic concerning AI acquisition, conceptualization, representation and knowledge application [4].

As discipline, it directs the task of building intelligent systems providing the tools and the methods that support the development of them. The key point of the development of a KBS is the moment to transfer the knowledge that the expert possesses to a real system. In this process they must not only capture the elements that compose the experts' domain, but rather one must also acquire the resolution methodologies that use these. The KE is mainly interested in the fact of "to discover" inside the intellectual universe of the human experts, all that is not written in rules and that they have been able to settle down through many years of work, of lived experiences and of failures. If the KE can also be defined as the task of to design and build Expert Systems (ES), a knowledge engineer is then the person that carries out all that is necessary to guarantee the success of a development of project of an ES; this includes the knowledge acquisition, the knowledge representation, the prototypes construction and the system construction [5].

2.1 Knowledge Acquisition and Knowledge Base

A Knowledge Acquisition (KA) methodology defines and guides the design of KA methods for particular application purposes. Knowledge elicitation denotes the initial steps of KA that identify or isolate and record the relevant expertise using one or multiple knowledge elicitation techniques. A KA method can involve a combination of several knowledge elicitation techniques which is then called knowledge elicitation strategy. There are several characteristics of KA that need to be considered when applying their methods, because it is a process of joint model building. The results of KA depend on the degree to which the knowledge engineer is familiar with the domain of the knowledge to be acquired and its later application. Also, it is noticed that the results of KA depend on the formalism that is used to represent the knowledge [6]. The sources are generally expert human but it can also be empiric data, books, cases of studies, etc. The required transformation to represent the expert knowledge in a program can be automated or partially automated in several ones.

General requirements exist for the automation of the KA and they should be considered before attempting this automation, such as independence of the domain and direct use of the experts without middlemen, multiple accesses to sources of such knowledge as text, interviews with experts and the experts' observations. Also support to diversity of perspectives including other experts, to diversity of types of knowledge and relationships among the knowledge, to the presentation of knowledge of diverse sources with clarity, in what refers to their derivation, consequences and structural relationships, to apply the knowledge to a variety domain and experience with their applications and to validation studies [7].

The automated methods for the KA include analogy, learning like apprentice, learning based on cases induction, decision trees analysis, discovery, learning based on explanations, neural nets, and the modification of rules, tools and helps, for the modeling and acquisition of knowledge that have been successful applied; they depend on intermediary representations constituting modeling languages of problems that help to fill the hole between the experts and the programs implementations. The AKE should be independent of the experts' domain, to be directly applicable for the experts without middleman able to ascend to knowledge sources (see figure 1).

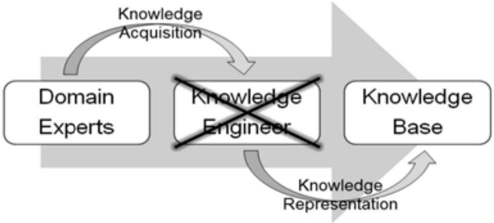


Fig. 1. Automated Knowledge Engineering

Diverse causes have taken to the construction of the Automated Knowledge Engineers (AKE); the descent in the cost of the software and hardware for ES; this has increased the demand of ES, greater than the quantity of AKE, and able to support ES [8]. The knowledge engineer's role, as middleman between the expert and the technology, sometimes is questioned. Not only because it increases the costs but also for their effectiveness, that is to say, it can get lost knowledge or it can influence subjectively on the Knowledge Base (KB) that is making. The automated knowledge acquisition keeps in mind in what measure belong together the description of the application domain that has the expert and the existent description in the KB and how to integrate the new information that the expert offers to the KB [9].

2.2 AKE to Acquire Individuals Mental Representation about Travel Behavior

While faced through complex choice problem like activity-travel option, persons generate a mental representation that allows them to understand the choice situation at hand and assess alternative courses of action. Mental representations include significant causal relations from realism as simplifications in people's mind. We have used for the capture of this data, in the knowledge engineering process, an AKE implementation where the user is able to select groups of variables depending of some categories, who characterize what they take into account in a daily travel activity.

There are diverse dialogues, trying to guide the user, but not in a strict way or order. In the software there are 32 different ways to sail from the beginning to the end, due to the flexibility that must always be in the data capture process, trying to adapt the Interface as much as possible to the user, guarantying then that the given information will be as natural and real as possible, and never forcing the user to give an answer or to fill a non-sense page. For each decision variable selected a matrix with attributes, situational and benefit variables exist, in this way respondents are asked to indicate the causal relations between the variables. This process is totally transparent to the user (that's way is called "Automated Knowledge Engineering").

In a case study 223 persons were already asked to use the software, and the results are considered good ones, given that the 99% of individuals were able to interact complete along with the AKE implementation, generating their own cognitive map about a shopping activity scenario that was given. Because of individual differences in the content of cognitive maps, there are different motivations or purposes for travel and different preferences for optimizing or satisfying decision strategies. Therefore human travel behavior is difficult to understand or predict.

3 Fuzzy Cognitive Maps as a knowledge Modeling Technique

FCM in a graphical illustration seem to be a signed directed graph with feedback, consisting of nodes and weighted arcs (see figure 2). Nodes of the graph place for the concepts that are used to express the system behavior, are connected by signed and weighted arcs representing the causal relationships that exist connecting the concepts.

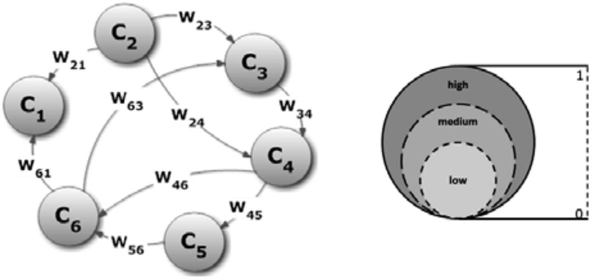


Fig. 2. Simple Fuzzy Cognitive Map. Concept Activation level.

The values in the graph are fuzzy, so concepts take values in the range between [0,1] and also the weights of the arcs are in the interval [-1,1]. The weights of the arcs between concept C_i and concept C_j could be positive ($W_{ij} > 0$) which means that an augment in the value of concept C_i leads to the increase of the value of concept C_j, and a decrease in the value of concept C_i leads to a reduce of the value of concept C_j. Or there is negative causality ($W_{ij} < 0$) which means that an increase in the value of concept C_i leads to the decrease of the value of concept C_j and vice versa [10].

Observing this graphical representation, it becomes clear which concept influences others, showing the interconnections between concepts and it permits updating in the construction of the graph. Each concept represents a characteristic of the system; in general it stands for events, actions, goals, trends of the system that is modeled as an FCM. Each concept is characterized by a number that represents its value and it results from the renovation of the real value of the system's variable. Beyond the graphical representation there is a mathematical model. It consists of a $1 \times n$ state vector A which includes the values of the n concepts and a $n \times n$ weight matrix W which gathers the weights W_{ij} of the interconnections between the n concepts [11].

The value of each concept is influenced by the values of the connected concepts with the appropriate weights and by its previous value. So the value A_i for each concept C_i is calculated by the following rule expressed in (1). 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_j and C_i, and *f* is a threshold function [12]. So the new state vector A_{new} is computed by multiplying the previous state vector A_{old} by the weight matrix W, see equation (2). The new vector shows the effect of the change in the value of one concept in the whole FCM. In order to build an FCM, the knowledge and experience of one expert on the system's operation must be used [13]. The expert determines the concepts that best illustrate the system; can be a feature of the system, a state, a variable, an input or an output of the system; identifying which factors are central for the modeling of the system and representing a concept for each one.

$$A_i = f\left(\sum_{\substack{j=1 \\ j \neq i}}^{n} A_j W_{ji}\right) \tag{1}$$

$$A_{new} = f(A_{old} \times W) \tag{2}$$

Moreover the expert has observed which elements of the system influence others; and for the corresponding concepts the expert determines the negative or positive effect of one concept on the others, with a fuzzy value for each interconnection, since it has been considered that there is a fuzzy degree of causation between concepts [14]. FCM are a powerful tool that can be used for modeling systems, avoiding many of the knowledge extraction problems which are usually present in rule based systems; and moreover it must be mentioned that cycles are allowed in the graph [15].

3.1 Tool Based on Fuzzy Cognitive Maps

The scientific literature reports some software developed for FCM modeling, as FCM Modeler [16] and FCM Designer [17]. The first one is a rustic incursion, while the second one is a better implementation, but with little experimental facilities. In figure 3 it is possible to observe the main window of our proposed tool, and a modeled example, in the interface appear some facilities to manage maps in general. From the data gathering described in section 2.2, it is possible to load automatically FCM structures, this tool is provided with a method that transform the KB extracted from individuals, into maps, so it is possible to simulate people behavior, but we have not always found a good correspondence between people decision and the predictions made by the maps, so it was necessary a reconfiguration of parameters. Therefore we developed an appropriate method described in next section.

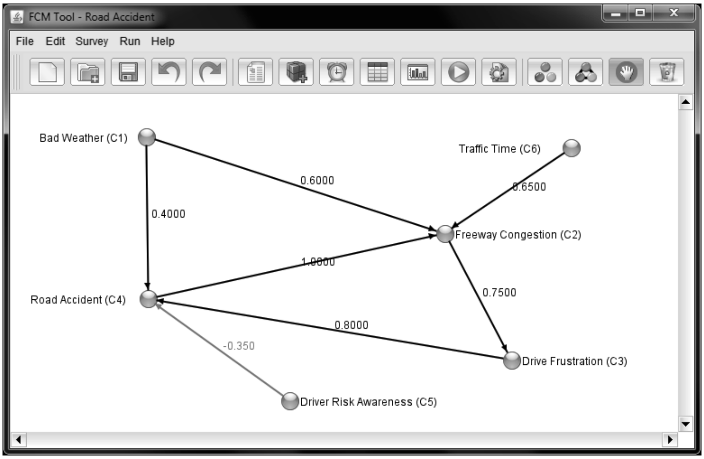


Fig. 3. Main view of the FCM Tool

4 Readjusting FCM Using a PSO Learning Method

Problems associated with development of FCM encourage researchers to work on automated or semi-automated computational methods for learning FCM structure 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 [18]. Researches on learning FCM models from data have resulted in a number of alternative approaches. One group of methods is aimed at providing a supplement tool that would help experts to develop accurate model based on their knowledge about a modeled system. Algorithms from the other group are oriented toward eliminating human from the entire development process, only historical data are necessary to establish FCM model [19].

Particle Swarm Optimization (PSO) method, which belongs to the class of Swarm Intelligence algorithms, can be used to learn FCM structure based on historical data, consisting in a sequence of state vectors that leads to a desired fixed-point attractor state. PSO is a population based algorithm, which goal is to perform a search by maintaining and transforming a population of individuals. This method improves the quality of resulting FCM model by minimizing an objective or heuristic function. The function incorporates human knowledge by adequate constraints, which guarantee that relationships within the model will retain the physical meaning defined [20].

The flow chart illustrated in figure 4 shows the main idea of the PSO application in the readjusting of the weight matrix, trying to find a better configuration that guaranty a convergence or waited results. PSO is applied straight forwardly using an objective function defined by the user. Each particle of the swarm is a weight matrix, encoded as a vector. First the concepts and relation are defined, and the construction of FCM is made, and then is possible to make simulations and obtain outputs due to the inference process.

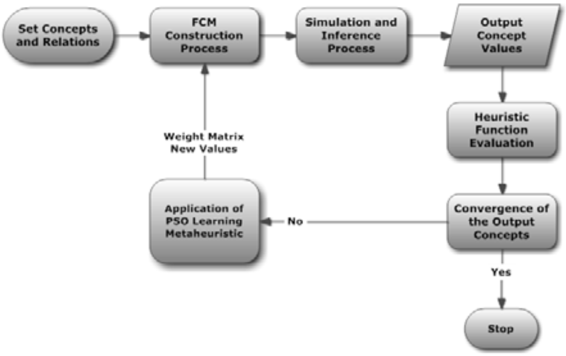
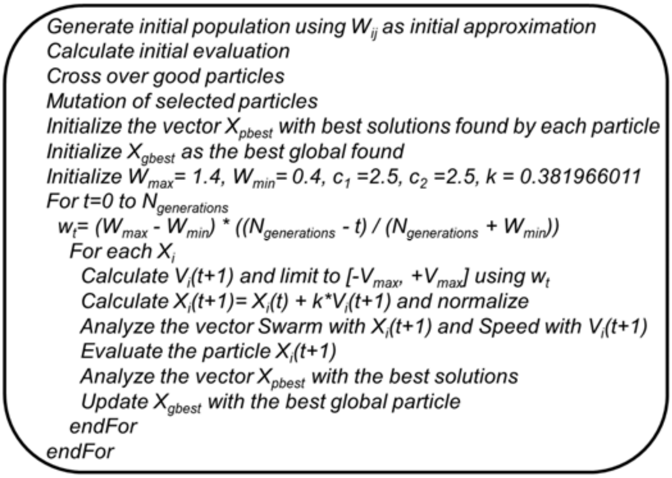

Set Concepts and Relations
FCM Construction Process
Simulation and Inference Process
Output Concept Values
Heuristic Function Evaluation
Convergence of the Output Concepts
Yes
Stop
No
Application of PSO Learning Metaheuristic
Weight Matrix New Values

Fig. 4. Using PSO for readjusting FCM

If the new values are not adequate, known by the execution of the heuristic function, then it is necessary a learning process (in this case through the use of PSO metaheuristic) having as results new values for the weight matrix. In the following pseudocode illustrated in figure 5, we can appreciate the general philosophy of our proposed method. In this case genetic algorithm operators are used as initial steps; mixed approaches have been performed so far. Using this approach, new zones of the search space are explored in a particular way, through the crossover of good initial particles and the mutation of some others, just to mention two possible ideas.

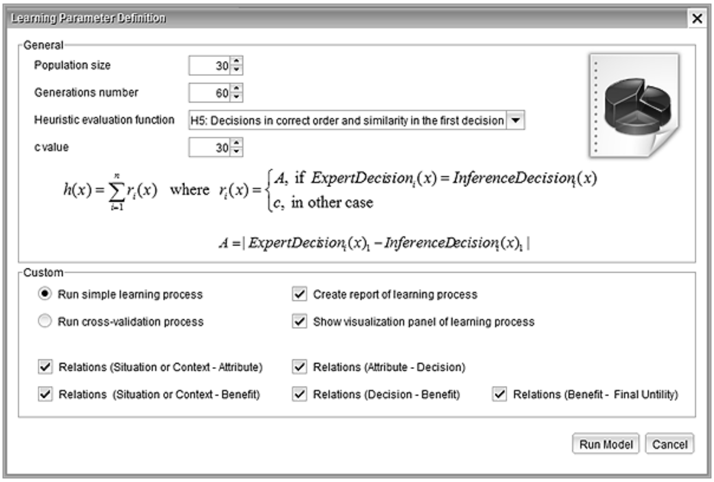


```
Generate initial population using $W_{ij}$ as initial approximation
Calculate initial evaluation
Cross over good particles
Mutation of selected particles
Initialize the vector $X_{pbest}$ with best solutions found by each particle
Initialize $X_{gbest}$ as the best global found
Initialize $W_{max}= 1.4$, $W_{min}= 0.4$, $c_1 =2.5$, $c_2 =2.5$, $k = 0.381966011$
For t=0 to $N_{generations}$
  $w_t= (W_{max} - W_{min}) * ((N_{generations} - t) / (N_{generations} + W_{min}))$
    For each $X_i$
      Calculate $V_i(t+1)$ and limit to $[-V_{max}, +V_{max}]$ using $w_t$
      Calculate $X_i(t+1)= X_i(t) + k*V_i(t+1)$ and normalize
      Analyze the vector Swarm with $X_i(t+1)$ and Speed with $V_i(t+1)$
      Evaluate the particle $X_i(t+1)$
      Analyze the vector $X_{pbest}$ with the best solutions
      Update $X_{gbest}$ with the best global particle
    endFor
endFor
```

Fig. 5. Pseudocode of the proposed method

4.1 Implementing the Learning Method Based on PSO for the FCM Readjustment

The necessary definition of parameters is done through the window shown in figure 6. In simulation and experiment in general, the visualization consists a fundamental aspect (that's why it was conceived a panel where the learning process can be observed, figure 7 shows an example). It is possible to see how the FCM is updated with a new weight matrix that better satisfy the waited results.



Learning Parameter Definition

General

Population size: 30

Generations number: 60

Heuristic evaluation function: H5: Decisions in correct order and similarity in the first decision

c value: 30

$$h(x)=\sum_{i=1}^{n} r_i(x) \quad \text{where} \quad r_i(x)=\begin{cases} A, \text{ if } ExpertDecision_i(x)=InferenceDecision_i(x) \\ c, \text{ in other case} \end{cases}$$

$$A=|\,ExpertDecision_i(x)_1 - InferenceDecision_i(x)_1\,|$$

Custom

● Run simple learning process

○ Run cross-validation process

☑ Create report of learning process

☑ Show visualization panel of learning process

☑ Relations (Situation or Context - Attribute)

☑ Relations (Attribute - Decision)

☑ Relations (Situation or Context - Benefit)

☑ Relations (Decision - Benefit)

☑ Relations (Benefit - Final Untility)

Fig. 6. Window for the PSO parameter specification

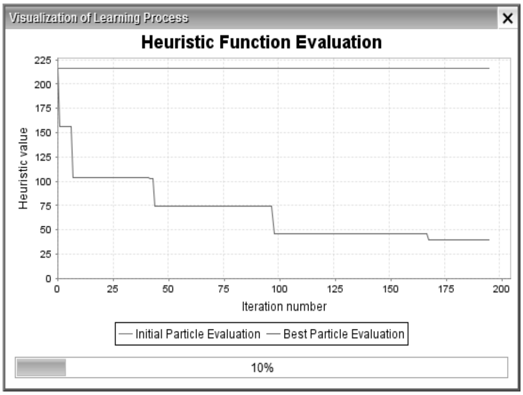


Fig. 7. Learning visualization panel

5 Validation

To the users participating in this research, virtual scenarios were presented, and the personal decisions were stored. Figure 8 shows the acting of the initial modeled FCM, for example, only the 24% of the maps were able to predict 100% scenarios. A FCM learning method, based on the PSO metaheuristic was applied, having the stored scenarios as training data, and the results show that after the learning process, 77% of maps were able to predict 100% of scenarios. It is considered a significant improvement over the maps, having structures able to simulate how people think when visiting the city center, specifically the transport mode they will use (car, bus or bike), offering policy makers, a tool to play with, to test new policies, and to know in advance possible resounding in society (buses cost, parking cost, bike incentive, etc.).

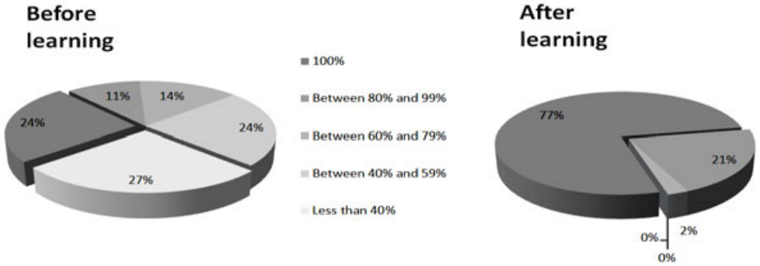


Fig. 8. Improving quality of knowledge structures

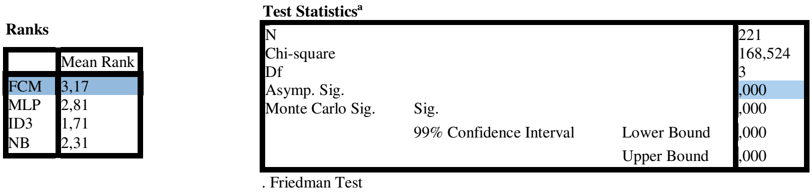
In Table 1 we detail the data organization for the statistic experiment, through a population comparison, to validate the performance of an FCM against other classical approaches such as Multilayer Perceptron (MLP), ID3 Decision Tree, or NaiveBayes (NB) classifier. The same knowledge had been modeled with these techniques. The idea consists in analyzing the possible significant difference among them using the classification percent (CP) they had obtained after an average of 3 times a cross-validation process with 10-folds.

Table 1. Data organization for processing

	FCM	**MLP**	**ID3**	**NB**
Expert 1	$CP_{FCM\ 1}$	$CP_{MLP\ 1}$	$CP_{ID3\ 1}$	$CP_{NB\ 1}$
Expert 2	$CP_{FCM\ 2}$	$CP_{MLP\ 2}$	$CP_{ID3\ 2}$	$CP_{NB\ 2}$
…	…	…	…	…
Expert 221	$CP_{FCM\ n}$	$CP_{MLP\ n}$	$CP_{ID3\ n}$	$CP_{NB\ n}$

After applying Kolmogorov-Smirnov test and having a non-normal distribution in our data, we apply non parametric Friedman test as shown in Table 2, where a signification less than 0.05 suggests to reject main hypothesis, therefor we can conclude that there exists a significant difference among groups. Looking to the mean ranks, the best value is given to FCM, however, it is not possible yet to affirm that our technique performs better than the others. Using a Wilcoxon test for related samples (see Table 3) it is possible to analyze per pairs, and in all cases the main hypothesis of the test is rejected and it is confirmed that there exists a significant difference between pairs. FCM definitely offer better results than the other approaches, and not only performed better, but also the most important is its capacity of presenting visual understanding information, combined with the classification skills, makes them seems a good approach for these kinds of tasks.

Table 2. Friedman Test to find significant differences



Ranks

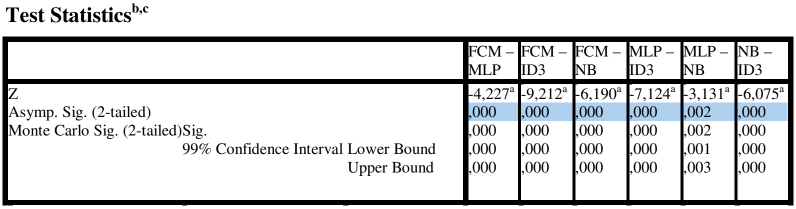
	Mean Rank
FCM	3,17
MLP	2,81
ID3	1,71
NB	2,31

Test Statistics[a]

N			221
Chi-square			168,524
Df			3
Asymp. Sig.			,000
Monte Carlo Sig.	Sig.		,000
	99% Confidence Interval	Lower Bound	,000
		Upper Bound	,000

. Friedman Test

Table 3. Wilcoxon Test for related samples



Test Statistics[b,c]

	FCM – MLP	FCM – ID3	FCM – NB	MLP – ID3	MLP – NB	NB – ID3
Z	-4,227[a]	-9,212[a]	-6,190[a]	-7,124[a]	-3,131[a]	-6,075[a]
Asymp. Sig. (2-tailed)	,000	,000	,000	,000	,002	,000
Monte Carlo Sig. (2-tailed)Sig.	,000	,000	,000	,000	,002	,000
99% Confidence Interval Lower Bound	,000	,000	,000	,000	,001	,000
Upper Bound	,000	,000	,000	,000	,003	,000

a. Based on negative ranks. b. Wilcoxon Signed Ranks Test
c. Based on 10000 sampled tables with starting seed 926214481.

Finally, Table 4 contains the average percentages after 3 times the same repeated experiment. First the learning scenarios serve for training, then for calculating optimistic estimation (resubstitution technique, empirical error) of the convergence. The resubstitution test is absolutely necessary because it reflects the self-consistency of the method, a prediction algorithm certainly cannot be deemed as a good one if its self-consistency is poor.

Table 4. Classification percent per technique, experiment and model

FCM	**MLP**	**ID3**	**NB**
FIRST DECISION			
Optimistic Model			
99.47	97.38	94.26	95.63
Pessimistic Model			
93.74	92.06	89.39	91.37
THREE DECISIONS			
Optimistic Model			
96.27	94.38	87.29	93.12
Pessimistic Model			
88.72	82.40	77.59	80.25

Later, the testing scenarios were used to obtain a pessimist estimation (cross-validation, real error) of the convergence through a cross-validation process with 10-folds. A cross-validation test for an independent testing data set is needed because it can reflect the effectiveness of the method in future practical application. The prediction capability had been measured in the forecast of the first possible decision and in three decisions given by the experts.

6 Conclusions

It has been examined Fuzzy Cognitive Maps as a theory used to model the behavior of complex systems, where is extremely difficult to describe the entire system by a precise mathematical model. Consequently, it is more attractive and practical to represent it in a graphical way showing the causal relationships between concepts.

A learning algorithm for determining a better weight matrix for the throughput of FCM was presented. An unsupervised weight adaptation methodology had been introduced to fine-tune FCM, contributing to the establishment of FCM as a robust technique. Experimental results based on simulations of the process system verify the effectiveness, validity and advantageous behavior of the proposed algorithm. The learned obtained FCM are still directly interpretable by humans and useful to extract information from data about the relations among concepts inside a certain domain.

The development of a tool based on FCM for the modeling of complex systems was presented, showing facilities for the creation of FCM, definition of parameters and options to make the inference process more comprehensible, understanding and used for simulation experiments. At the end, a real case study was presented, showing a possible Travel Behavior modeling through FCM, and the benefits of the application of a learning method inspired in the PSO metaheuristic, obtaining an improvement on the knowledge structures originally modeled. In this shown example a social and politic repercussion is evident, as we offer to policymakers a framework and real data to play with, in order to study and simulate individual behavior and produce important knowledge to use in the development of city infrastructure and demographic planning.

References

1. Gutiérrez, J.: Análisis de los efectos de las infraestructuras de transporte sobre la accesibilidad y la cohesión regional. Estudios de Construcción y Transportes. Ministerio Español de Fomento (2006)
2. Janssens, D.: The presentation of an activity-based approach for surveying and modelling travel behaviour, Tweede Belgische Geografendag (2005)
3. Janssens, D.: Tracking Down the Effects of Travel Demand Policies. Urbanism on Track. Research in Urbanism Series. IOS Press (2008)
4. Cassin, P.: Ontology Extraction for Educational Knowledge Bases. In: Spring Symposium on Agent Mediated Knowledge Management. Stanford University, American Association of Artificial Intelligence (2003)
5. Mostow, J.: Some useful tactics to modify, map and mine data from intelligent tutors. Natural Language Engineering 12, 195–208 (2006)
6. Rosé, C.: Overcoming the knowledge engineering bottleneck for understanding student language input. In: International Conference of Artificial Intelligence and Education (2003)

7. Soller, A.: Knowledge acquisition for adaptive collaborative learning environments. In: American Association for Artificial Intelligence Fall Symposium. AAAI Press (2000)
8. Woolf, B.: Knowledge-based Training Systems and the Engineering of Instruction. Macmillan Reference, 339–357 (2000)
9. León, M.: A Revision and Experience using Cognitive Mapping and Knowledge Engineering in Travel Behavior Sciences. Polibits 42, 43–49 (2010)
10. Kosko, B.: Neural Networks and Fuzzy systems, a dynamic system approach to machine intelligence, p. 244. Prentice-Hall, Englewood Cliffs (1992)
11. Parpola, P.: Inference in the SOOKAT object-oriented knowledge acquisition tool. Knowledge and Information Systems (2005)
12. Kosko, B.: Fuzzy Cognitive Maps. International Journal of Man-Machine Studies 24, 65–75 (1986)
13. Koulouritios, D.: Efficiently Modeling and Controlling Complex Dynamic Systems using Evolutionary Fuzzy Cognitive Maps. International Journal of Computational Cognition 1, 41–65 (2003)
14. Wei, Z.: Using fuzzy cognitive time maps for modeling and evaluating trust dynamics in the virtual enterprises. Expert Systems with Applications, 1583–1592 (2008)
15. Xirogiannis, G.: Fuzzy Cognitive Maps as a Back End to Knowledge-based Systems in Geographically Dispersed Financial Organizations. Knowledge and Process Management 11, 137–154 (2004)
16. Mohr, S.: Software Design for a Fuzzy Cognitive Map Modeling Tool. Tensselaer Polytechnic Institute (1997)
17. Aguilar, J.: A Dynamic Fuzzy-Cognitive-Map Approach Based on Random Neural Networks. Journal of Computational Cognition 1, 91–107 (2003)
18. Mcmichael, J.: Optimizing Fuzzy Cognitive Maps with a Genetic Algorithm AIAA 1st Intelligent Systems Technical Conference (2004)
19. González, J.: A cognitive map and fuzzy inference engine model for online design and self-fine-tuning of fuzzy logic controllers. Int. J. Intell. Syst. 24(11), 1134–1173 (2009)
20. Stach, W.: Genetic learning of fuzzy cognitive maps. Fuzzy Sets and Systems archive 153(3) (2005)

Evaluating Probabilistic Models Learned from Data

Pablo H. Ibargüengoytia, Miguel A. Delgadillo, and Uriel A. García

Instituto de Investigaciones Eléctricas
Av. Reforma 113, Palmira
Cuernavaca, Mor., 62490, México
{pibar,madv,uriel.garcia}@iie.org.mx

Abstract. Several learning algorithms have been proposed to construct probabilistic models from data using the Bayesian networks mechanism. Some of them permit the participation of human experts in order to create a knowledge representation of the domain. However, multiple different models may result for the same problem using the same data set. This paper presents the experiences in the construction of a probabilistic model that conforms a viscosity virtual sensor. Several experiments have been conduced and several different models have been obtained. This paper describes the evaluation implemented of all models under different criteria. The analysis of the models and the conclusions identified are included in this paper.

Keywords: Bayesian networks, Learning algorithms, Model evaluation, Virtual sensors.

1 Introduction

In the present days, the automation of human activities is increasing due to the availability of hardware, software and sensors for all kind of applications. This fact produces the acquisition of great amounts of data. Consider for example each transaction with a credit card or each item purchased in a shopping center.

This automatic data acquisition represents a challenge for the Artificial Intelligence (AI) techniques. The challenge of knowledge discovering in data bases.

This paper deals with the problem of generating the best possible probabilistic model that conforms a viscosity virtual sensor for controlling the combustion in a thermoelectrical power plant.

The viscosity virtual sensor [4] consists in the on-line estimation of the viscosity of fuel oil of a thermoelectric power plant. This estimation is based on probabilistic models constructed from data acquired from the power plant. The data is formed by the value of several variables related to the combustion of the fuel oil in the plant.

Viscosity is a property of the fuel oil and it is important to measure for the combustion control. One option is the use of hardware viscosity meters. However, they are expensive and difficult to operate under plant operating conditions, and

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 95–106, 2011.

to maintain. The second option to measure the viscosity is chemical analysis in laboratory. This option is used every time a new supply of fuel arrives to a power plant. However, this procedure is off line and takes more than an hour to obtain the measured result. The third option is the development of a viscosity virtual sensor that *estimates* the value of the viscosity given related measurements.

This paper describes the automatic learning process that was followed in order to define the best model for the viscosity virtual sensor.

This paper is organized as follows. The next section briefly explains the application domain where this work is conduced, namely the construction of a viscosity virtual sensor. Next, section 3 describes the different tools developed to evaluate probabilistic models. Section 4 exposes the set of experiments developed, their evaluation and the discussion of the results. Finally, section 5 concludes the paper and suggest the future work in this project.

2 Viscosity Virtual Sensor

The power generation can be summarized as follows. Water is heated in huge boilers that produce saturated steam that feeds turbines that are coupled to electric generators. The calorific energy of the steam is transformed in mechanical work at the turbines and this work is transformed to electricity in the generators. While more gas is consumed in the boiler, more steam is produced and hence, more power is generated. This generation process is measured by an index called thermal regime. This index relates the Mega Watts produced per liter of oil burned.

To increase the thermal regime index, the combustion in the boiler is an important process to control. Usually, control systems regulate the aperture of the fuel oil valve to feed more or less oil to the burners in the boiler. However, the optimal combustion depends on several factors. One of these factors is the oil atomization at the burners. If the droplet of oil is too big, only a small portion of it will be burned and the rest is expelled through contaminant smoke. If the droplet is too small, the reaction of fuel and oxygen is incomplete and produces also contaminant residues and low combustion performance. Thus, in order to have a good oil atomization, an exact viscosity is required in the flow of oil to the burners. Viscosity is a property of the matter that offers resistance to flow. The viscosity changes mainly with the temperature. Thus, an optimal combustion control includes the determination of the viscosity of the input oil and its optimal heating, so the viscosity can be driven to the required value. This produces a good combustion that generates steam for power generation. Fossil oil is provided to the electric generation plants from different sources and different qualities.

The virtual sensor design starts in the selection and acquisition of the related signals from the plant. The hypothesis is that, the related signals may be generated from: before, during and after the combustion. In the selected plant there is a hardware viscosity meter that is used to compare the process signals with the measured viscosity. Thus, a huge historical data set is acquired with measures every 5 seconds during several days. In a first attempt, several variables were sampled. This represented an enormous number of measurements.

The data set is cleaned, normalized and discretized to be ready for the learning algorithms. With the learning algorithm, a probabilistic model is constructed, based on Bayesian networks[7]. The probabilistic model is later utilized in the virtual sensor software. This computer program opens the models and reads real time data. The model infers the viscosity value and calculates the final viscosity value.

The first problem was the selection of related signals. Pipe and instrumentation diagrams (PIDs) were provided and a full explanation of the performance of units 1 and 2 of the Tuxpan Thermoelectric power plant, operated by the Federal Commission of Electricity (CFE) in Mexico.

Tuxpan is a 6 units power plant located at north of the state of Veracruz, in the Gulf of Mexico littoral. This plant was selected since it has installed viscosity meters in units 1 and 2. This instrument is needed to acquire historical data including the viscosity in order to find the relationship between viscosity and other signals.

In a first attempt, 32 variables were selected for constructing the probabilistic model. However, revising the behavior of each signal with respect to the viscosity, and consulting the experts in combustion, only a few variables remain. Table 1 describes the ID and the description of the variables selected. Besides the variables extracted from the power plant data base, some other were calculated. The first one is the Thermal rating (Rt) and air-fuel ratio (Rac). Thermal rating reflects the performance of the boiler since it relates the energy balance, i.e., the watts produced by fuel unit.

Data from 32 variables every 5 seconds were solicited to the plant personnel from several days. One day before a change of fuel, the day of change and the day after the change. However, the first problem was to deal with this huge amount of data. There are more than 17,000 registers per day.

There exist several learning algorithms that construct the structure of the network, and calculate the numerical parameters. The selection of the correct algorithm depends on several criteria. For example, it is required the participation of human experts in the definition of the probabilistic dependencies. Also, it is required the construction of models with relatively low interconnection. This is because the virtual sensor works on-line, i.e., the probabilistic inference must be calculated fast enough.

The first problem in the learning process for the viscosity virtual sensor is the selection of the signals that may be related to the viscosity. From these signals, an historical file with the signals every 5 seconds was obtained. This means more than 17,000 samples from 34 variables. However, this resulted in an impractical amount of information. A selection of attributes was required. This variable selection process was carried out with experts advice and attribute selection algorithms from weka package [3]. Table 1 describes the final set of variables, their identification and their description.

Table 1. Set of variables selected to construct the model

ID	Name	Description
T462	U2BAT462	Internal boiler temperature
A1000	U2JDA1000	Fuel viscosity
F592	U2JDF592	Fuel total flow to burners
P600	U2JDP600	Fuel pressure after the fuel valve
P635	U2JDP635	Atomization steam pressure
T590	U2JDT590	Fuel temperature
A470Y	U2BAA470Y	Oxygen in the combustion gases
W01	U2GH31W01	Power generated
F457	U2BAF457	Air total flow to combustion
Z635	U2JDZ635	Aperture of the atomization steam valve
Rt	Rt	Thermal rating
Rac	Rac	Fuel - air ratio

3 Evaluation Tools

The evaluation of models conduced in this work represents different characteristics.

- In some experiments, we try to evaluate the learning of a probabilistic model using historical data.
- Other experiments evaluate the construction of a Bayesian network with different causality assumed between the variables.
- Other kind of experiments evaluates the performance of the same model but changing some characteristics of the data. For example, inserting delays or increasing the number of intervals in the discretization.
- The last experiments evaluate the participation of certain variables in the estimation process.

Since these experiments are different, we need different evaluation tools. This section describes some tools for evaluation. In some cases, we used basic tools like Cross-validation [6]. Some other basic techniques include ROC curves [8] that depict the performance of a classifier plotting number of positive against the number of negatives. However, our problem can not be expressed as true or false case, but the accuracy of the viscosity estimation.

The power plant personnel provided 17 days of information from 2009 and 2010. The selection of the days responds to the change of fuel supplier. In Tuxpan, fuel oil can be supplied by the Mexican oil company Pemex or imported. The national fuel is usually the last sub product of the oil refining and use to be of low quality with a high viscosity. On the other hand, imported fuel usually is of high quality and low viscosity. Both kind of fuel are mixed in the inlet tank and this mixture results in a fuel with unknown characteristics. Thus, data from different kind of fuel oil, and different operational conditions result in 270,000 registers of 14 variables. With this huge amount of information, a Bayesian network structure, and parameters were necessary to relate all the variables with the viscosity measure from the hardware viscosity meter. The K2 [2] learning algorithm was used. The K2 algorithm allows the expert user to indicate a known causal relation between the variables. For example, it is certainly known that a change in fuel temperature causes a change in fuel viscosity. Besides, K2 restricts the number of parents that a node may have. This is important to keep low interconnection between nodes and hence, to maintain low computational cost in the inference. Five different criteria are involved in the model learning that have to be defined:

1. Selection of the set of variables that influences the estimation of viscosity. They can be from before, during or after the combustion.
2. Processing of some variables according to their participation in the combustion. For example, some delay is needed in variables after the combustion to compare with variables from before the combustion.
3. Normalization of all variables values to a value between 0 and 1. This allows to comparing the behavior of all variables together.
4. Number of intervals in the discretization of continuous variables.
5. Causal relation between variables. This is the parameter that K2 needs to create the structure. It is indicated in the order of the columns in the data table.

The combination of these criteria results in a large number of valid models that may produce different results. The challenge is to recognize the best combination of criteria that produces the best model and consequently, the best viscosity estimation. The learning procedure followed in this project was the construction of the model utilizing the K2 algorithm considering a specific set of criteria. For example, discretizing all variables in ten intervals without any delay. Next, a specific data set for testing was used to evaluate the performance of the model. The tools available for this evaluation are described next.

3.1 Bayesian Information Criterion or BIC Score

Given that the models are constructed with real time data from the plant, and since there can be different causality considerations for the learning algorithm, to measure how well the resulting model represents the data is required. One common measure is the Bayesian information criterion, or BIC score [2]. The mathematical definition of the BIC score is:

$$BIC = n \cdot ln(\sigma_e^2) + k \cdot ln(n)$$

Where n is the number of data registers in the learning data set, k is the number of free parameters to estimate and σ_e^2 is the error variance defined as:

$$\sigma_e^2 = \frac{1}{n}\sum_{1}^{n}(x_i - \overline{x})^2$$

Thus, obtaining different models with from different criteria, the model with the higher value of BIC is the one to be preferred. Notice that BIC score for discrete variables is always negative. Thus, the lower negative value (the higher BIC value) is the preferred model. Section 4 presents the experimental results.

3.2 Data Conflict Analysis

Given that the models are constructed with real time data from the plant, and given that not all the state space is explored, some conflicts arise in the testing phase. The data conflict analysis detects when rare or invalid evidence is received. Given a set of observations or evidence $e = \{e_1, e_2, \ldots, e_n\}$ the conflict is defined as [5]:

$$Conf(e) = log\frac{\prod_{i=1}^{n} P(e_i)}{P(e)}$$

The conflict can be calculated after data from the variables is loaded in the model and new viscosity estimation is obtained. In other words, new evidence is loaded. Thus, if conflict $conf(e)$ is positive, then there exist a negative correlation between the related variables' values and a conflict is detected. On the contrary, if $conf(e) < 0$, then the evidence is presumably consistent with the model. Some experiments were conducted and some conflicts were detected. Section 4 presents the experimental results.

3.3 Parameter Sensitivity Analysis

Given a learned model, when revising the viscosity estimation, some unexpected values have been obtained. Sometimes the estimation can be very sensible to variations in one or more evidence variables. The parameter sensibility analysis [1] is a function that describes the sensitivity on the hypothesis variable, i.e., the viscosity, to changes on the value of the some related variable, e.g. the fuel temperature. It is used to test for example, if the number of intervals in the discretization of one variable is appropriate for the viscosity estimation given the data set. Section 4 presents the experimental results.

4 Learning Experiments

The first problem in this learning process was the selection of the set of variables that are measured directly by the control system, and may be related to the viscosity. The variables are generated from before, during or after the combustion. This selection was defined with multiple interviews with experts in combustion. Some selected variables are generated before combustion like flow of fuel to burners or fuel temperature. Other variables are generated during the combustion like internal boiler temperature, and other are generated after the combustion like the generated power. The result of this process was a large set of variables. However, some of them were discarded by the K2 learning algorithm. These variables were isolated in the model from the rest of variables. The resulting set is indicated in table 1.

The basic description of the experiments is the following. Given a set of data, the K2 algorithm is applied and a Bayesian network is obtained. For example, the network of Fig.2. Next, we introduce the network in our software together with the testing data set and compare the viscosity estimation based on probabilistic propagation, and the viscosity from the hardware meter. An error is calculated and reported in the experiment. However, several combination of characteristics of the experiments are possible. For example, inserting delays or not, discretizing with 10 or 20 intervals, normalizing or not.

Notice that the number of combinations of characteristics grows exponentially. We only tested the change of one characteristic for each experiment, assuming that the effects of each characteristic is independent from the others. The experiments conduced are described next.

4.1 Experiments Revising Performance

The first set of experiments were planned to define aspects of the learning process like order in the variables, normalization, discretization and delays. Table 2 describes these experiments. The first column identifies the experiment. The second column describes the learning aspect to test, e.g., different number of intervals in the discretization. The results columns indicate the average of error and standard deviation between all the estimations. For example, if we use one day of information for testing, and we obtain variables values every 5 seconds, the number of estimations is above 17,000 tests. Finally, an indication of the error method is included.

Table 2. Description of the experiments 1 to 3. Generating different models.

Exp.	**Object of the test**	**Results**	
		Avrge	StdDev
1	Use of all data available separating data for training and testing	5.78	4.58
2	Same as exp. 1 with delay in corresponding variables	5.64	4.72
3	Same as exp 2 but using discretization in 20 intervals in all variables	2.63	3.2

The evaluation parameter for these experiments was the error found between the estimated viscosity and the measure viscosity using the viscosity meter, i.e., $Error = \frac{(V_{read} - V_{est})}{(V_{read})} \times 100$. This measure represents the performance of the model for the estimation of the viscosity. Notice that there was a decrement in the average error when a delay was inserted in some variables. Also, another significant decrement when 20 intervals were used for the discretization. This fact is expected since discretizing a continuous value variable necessarily inserts an error in the processing.

Table 3. Description of the experiments 4 to 8. Same model, different characteristics.

Exp.	Object of the test	Results	
		Avrge	StdDev
4	Use of new data from the plant for testing	9.1	6.38
5	Use of all data for training but excluding one day data for testing. Use of filter in the estimated viscosity	3.59	3.61
6	Same as exp. 5 but using a delay in the training data	4.45	4.75
7	Same as exp. 5 but excluding evidence in Z635	4.65	5.33
8	Same as exp. 5 but excluding variable Z635. Use of filter in the estimation	4.57	5.22

The second set of experiments were planned to evaluate the current defined model with new real data received from plant. We used the received data to test. We used also a new error measure. Table 3 describes these experiments.

The evaluation of the models in this case was also the comparison between the error in the estimation. However, the error was calculated differently in these experiments. Now, error was defined as $Error = \frac{(V_{read} - V_{est})}{(V_{max} - V_{min})} \times 100$ where V_{max} and V_{min} represent the span of values where the viscosity meter was calibrated. This is the normal error measure in a normal instrument according to the instrumentists experts. Notice that experiment 4 shows a high error average. This is because the new data was taken from a diverse operational condition of the plant. Next, we integrated all the data available and separate data sets for training and testing. In experiment 5 we used a filter in the estimated viscosity signal. Experiment 6 was conducted using delay in both, training data and testing data. Experiments 7 and 8 were utilized to identify the participation of variable Z635 in the estimation. It resulted that the use of this variable produces high positive conflict when propagating probabilities. In fact, we decide to exclude this variable in the following models.

The third set of experiments were planned to evaluate the models with respect to the BIC score explained above. We use the complete data set obtained from plant for training the model and calculating the BIC score. Additionally, we run experiments to check the error average. Table 4 describes these experiments.

In experiments 9 and 10, we compare the model score without (exp. 9) and with (exp. 10) delays in the corresponding variables. Next, in experiments 11 to 13, we found an error in part of the data set and exclude this data in the training set. We discover that three days of information were taken from unit 1 of the plant, instead of unit 2. Experiment 11 shows the model using the correct data set, normalized, using delay in the corresponding variables and 20 intervals. We use the order A of variables for the K2 algorithm shown in Table 5. In experiment 12 we use exactly the opposite order as shown in line B of Table 5 and experiment 13 with random order as shown in line C. Finally, experiment 14 shows the experiment using a manual discretization. Notice that the model of exp. 11 obtained the best BIC score as expected.

Table 4. Description of the experiments 9 to 14. Evaluating Bayesian networks when human knowledge is integrated.

Exp.	Object of the test	Results		BIC
		Avrge	StdDv	
9	Use of the 17 files for training. It results in the same structure than exp. 8. No delay was applied in corresponding variables	2.38	2.42	-3,677,110
10	Same as exp. 9 but using delay in corresponding variables	2.5	2.09	-1,461,010
11	Excluding files from 2009. They are from Unit 1. 20 intervals, normalized with delay	1.64	1.82	-2,848,280
12	Experiment 11 with an order of variables exactly opposite. Order B in Table 5	1.46	1.74	-2,936,160
13	Experiment 11 with an order of variables exactly opposite. Order C in Table 5			-2,908,010

Table 5. Order of variables for K2 algorithm

	Order of variables for K2											
A	T590	F592	A1000	P600	F457	P635	Z635	Rac	T462	Rt	W01	A470Y
B	A470Y	W01	Rt	T462	Rac	Z635	P635	F457	P600	A1000	F592	T590
C	W01	F457	F592	Rt	P600	T590	A470Y	P635	T462	A1000	Rac	

4.2 Revising Markov Blankets

Besides the scores obtained in the design of the models, we are interested in the definition of the set of variables that allows estimating the viscosity on-line. For experts in combustion, this is the main contribution of this project. Figure 1 shows the variables that belong to the Markov blanket (MB) of viscosity (node A1000) in every model obtained in the experiments.

Notice that the generation and the air/fuel ratio (variables W01 and Rac) never form part of the MB o A1000. Also notice that variable Z635 was eliminated from the models. Summarizing, the variable set that are related with the viscosity is formed by the following set:

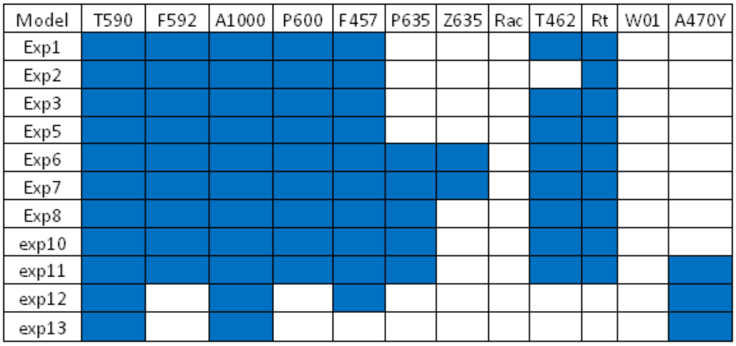
$$\{T590, F592, P600, F457, P635, T462 \text{ and } Rt\}$$

4.3 Defined Best Model to Estimate the Viscosity

After conducting all the experiments, a definite model is selected. Figure 2 shows this model.

Additionally, the following considerations are concluded for the generation of the best model:

1. Use of normalization of the data set,
2. Apply delay in the corresponding variables,



Model	T590	F592	A1000	P600	F457	P635	Z635	Rac	T462	Rt	W01	A470Y
Exp1	■	■	■	■	■				■	■		
Exp2	■	■	■	■	■					■		
Exp3	■	■	■	■	■				■	■		
Exp5	■	■	■	■	■				■	■		
Exp6	■	■	■	■	■	■	■		■	■		
Exp7	■	■	■	■	■	■	■		■	■		
Exp8	■	■	■	■	■	■			■	■		
exp10	■	■	■	■	■	■			■	■		
exp11	■	■	■	■	■	■			■	■		■
exp12	■		■		■							■
exp13	■		■									■

Fig. 1. Markov blankets of all models constructed in the experiments

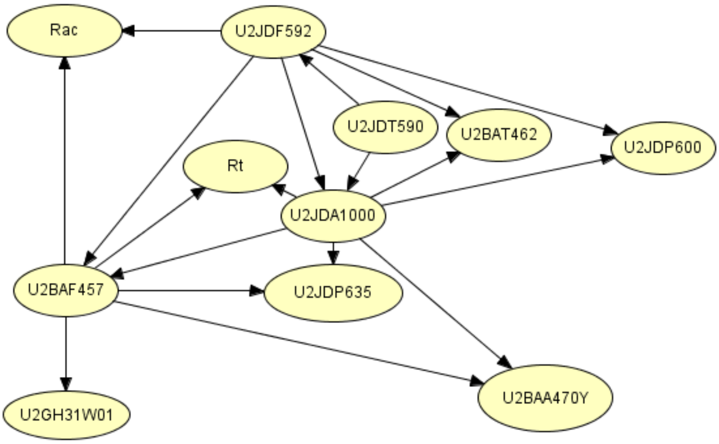


Fig. 2. Definitive Bayesian network of the viscosity virtual sensor

3. Use the order A of Table 5, and
4. Discretize using 20 intervals in most of the variables.

Figure 3 shows a sample of the results obtained in the final experiments considering all the learned considerations described above.

Vest and *Vread* corresponds to the estimated viscosity and the measured viscosity by the viscosity meter. Error graph is the resulting error in the estimation with respect to the range of the physical instrument. Horizontal axis represents the time, where the numbers represent the samples of the estimation in the experiment. The first graph show results from the sample 1000 to 8000. Vertical axis represents the normalized value of the viscosities, and the percentage of the error signal.

Notice that the estimated signal always follows the measured signal. However, there exist some instances where the estimated signal presents some deviations

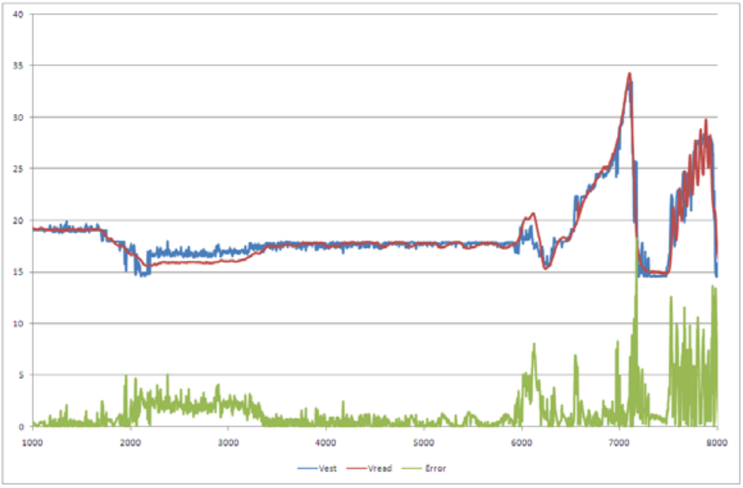


Fig. 3. Comparison between estimated and measured viscosities, and the error produced

that increase the error. Future experiments will improve the model and the treatment of the signals.

5 Conclusions and Future Work

This project started with the availability of high volume of historical data including the viscosity measure by a hardware viscosity meter, and with the hypothesis that the viscosity can be inferred on-line using common variables. Thus, the main activity in this project is the learning procedure followed to obtain the best possible model for estimating the fuel oil viscosity.

It has been shown what specific set of variables is enough for estimating viscosity from measurements on line. It has also been shown that some delay in necessary in the variables that are retarded on the combustion. Normalization is necessary in order to compare the behavior of all signals together. Finally, we found the order of the variables according to their causal relation between the combustion process.

The immediate future work is the installation and evaluation of the viscosity virtual sensor in the Tuxpan Power plant. New data will permit to improve the models for better estimation. Also, different kind of models can be used to compare their performance, for example the use of Bayesian classifiers.

On the plant, the final evaluation will be on the effects of calculating the viscosity in the control of the combustion. This means, obtaining the current viscosity allows calculating the ideal fuel oil temperature that produces the optimal atomization and hence, an optimal combustion. The power generation will be more efficient and cleaner with the environment.

Acknowledgment. This work has been supported by the Instituto de Investigaciones Eléctricas-México (project no. 13665) and the Sectorial Found Conacyt-CFE, project no. 89104.

References

1. Andersen, S.K., Olesen, K.G., Jensen, F.V., Jensen, F.: Hugin: a shell for building bayesian belief universes for expert systems. In: Proc. Eleventh Joint Conference on Artificial Intelligence, IJCAI, Detroit, Michigan, U.S.A, August 20-25, pp. 1080–1085 (1989)
2. Cooper, G.F., Herskovitz, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9(4), 309–348 (1992)
3. 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)
4. Ibargüengoytia, P.H., Delgadillo, M.A.: On-line viscosity virtual sensor for optimizing the combustion in power plants. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 463–472. Springer, Heidelberg (2010)
5. Jensen, F.V., Chamberlain, B., Nordahl, T., Jensen, F.: Analysis in hugin of data conflict. In: Bonissone, P.P., Henrion, M., Kanal, L.N., Lemmer, J.F. (eds.) Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence (UAI 1991), vol. 6, pp. 519–528. Elsevier Science Publishers, Amsterdam (1991)
6. Kohavi, R.: A study of cross-validation and boostrap for accuracy estimation and model selection. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 1137–1143. Morgan Kaufmann, San Francisco (1995)
7. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
8. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparision under imprecise class and cost distributions. In: Hekerman, D., Mannila, H., Pregibon, D., Uthurusamy, R. (eds.) Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press, Huntington Beach (1997)

A Mutation-Selection Algorithm for the Problem of Minimum Brauer Chains

Arturo Rodriguez-Cristerna, José Torres-Jiménez, Ivan Rivera-Islas,
Cindy G. Hernandez-Morales, Hillel Romero-Monsivais, and Adan Jose-Garcia

Information Technology Laboratory, CINVESTAV-Tamaulipas Km. 5.5 Carretera Cd.
Victoria-Soto la Marina, 87130, Cd. Victoria Tamps., Mexico
arodriguez@tamps.cinvestav.mx, jtj@cinvestav.mx
{rrivera,chernandez,hromero,ajose}@tamps.cinvestav.mx

Abstract. This paper aims to face the problem of getting Brauer Chains (BC) of minimum length by using a Mutation-Selection (MS) algorithm and a representation based on the Factorial Number System (FNS). We explain our MS strategy and report the experimental results for a benchmark considered difficult to show that this approach is a viable alternative to solve this problem by getting the shortest BCs reported in the literature and in a reasonable time. Also, it was used a fine-tuning process for the MS algorithm, which was done with the help of Covering Arrays (CA) and the solutions of a Diophantine Equation (DE).

Keywords: Brauer chain, Mutation-Selection, Factorial Number System, Covering Arrays, Diophantine Equation.

1 Introduction

An addition chain for a positive integer n is

Definition 1. *A set* $1{=}a_0 < a_1 < ... < a_r = n$ *of integers such that for each* $i \geq 1$, $a_i = a_j + a_k$ *for some* $k \leq j < i$.

The length in the addition chain s is denoted by $l(s)$ and its equal to r. Here, every set $\{j, k\}$ in an addition chain is called step, and according with the types of values of j and k along the chain, it takes some particular name. For our purpose we are going to use j as $i-1$, which is called a star step, and "an addition chain that consists entirely of star steps is called a star chain" [13] or Brauer Chain (BC) [7] in honor of the definition that Brauer gives in [2]. Where a BC C has the smallest length r for a number n, we can say that C is a Minimum Brauer Chain (MBC) for n. The search space for constructing a BC for the number n is $r!$ and can be described as a tree, where every non root node is made by a star step. We can see this space in the **Figure 1**, also it is observed two ways to form a MBC for $n = 6$, where the first one is *1, 2, 3, 6* and the second one is *1, 2, 4, 6*. One of the principal uses of Minimum Addition Chains (MAC) is in the reduction of the steps in a modular exponentiation (repetition of modular multiplications), which is an important operation during data coding in cryptosystems as RSA

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 107–118, 2011.

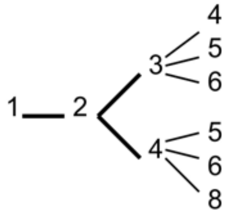


Fig. 1. Search space of a BC with length $r = 3$

encryption scheme [10]. This is because the cost of the multiplications required to produce an exponentiation is high, then, reducing the number of steps in a modular exponentiation improves the performance and impacts in the efficiency of a cryptosystem [1]. The searching process for a MBC for numbers like 7 or 23 is relatively easy, but for the number 14143037, is not because the search space becomes very large. In this paper we propose a MS algorithm to face the problem of getting MBCs, which uses a representation based on the Factorial Number System (FNS). The remaining of this paper is organized as follows. **Section 2** describes the relevant related work of our research, **Section 3** gives the details of our proposed approach, **Section 4** explain the experimental design and the fine-tuning process followed, **Section 5** shows the results obtained and finally **Section 6** gives the reached conclusions.

2 Relevant Related Work

Thurber (1999) explored an algorithm to generate MACs based on a backtracking algorithm using branch and bound methods. The representation used is a tree of k levels that explores a search space of size at least $k!$.

Nedja and Moruelle (2002) gave an approach based in the $m-ary$ method using a parallel implementation to compute MACs, by decomposing the exponent in blocks (also called windows) containing successive digits. Their strategy produces variable length zero-partitions and one-partitions, using a lower number of operations than the binary method.

Other methodology explored by Nedja and Moruelle (2003) uses large windows inside a genetic algorithm using a binary encoding.

Bleichenbacher and Flammenkamp (1997) produced MACs by using direct acyclic graphs and a backtrack search. They also use an optimization stage inside their approach where special cases of addition chains are checked and replaced with another equivalent chain in order to get a smaller search tree.

Gelgi and Onus (2006), proposed some heuristics approaches to do an approximation for the problem of get an MBC. They present five approaches: the first three set the index positions 3 and 4 of the BC with the numbers $(\mathit{3}, \mathit{5}), (\mathit{3}, \mathit{6})$ or $(\mathit{4}, \mathit{8})$, the fourth approach is a factorization heuristic and the fifth approach is a heuristic based on dynamic programming that uses previous solutions to obtain a better one. They found empirically, that their dynamic heuristic approach has an approximation ratio (obtained length / minimum length) of 1.1 with $0 \leq n \leq 20000$.

3 Proposed Approach

3.1 Mutation-Selection Algorithm

In order to present the mutation algorithm used, a brief description of how it works is given. Assuming that $f(x)$ is an objective function and x belongs to a definite and bounded realm, the search space is the set of values that can take the variable x. A trial is an evaluation of $f(x)$ for a specific value, and it is done trying to find an optimal x value.

A Mutation-Selection (MS) algorithm, uses one or more points in the search space, called *parent-points*, to generate multiple points through the use of mutation operators, these generated points are called *children-points*, subsequently *children-points* are evaluated in search of an optimal point. If no optimal point is found, it comes the stage of selecting the members of the next generation of new parents, which is called a survivor selection, and all the process is done over again. This cycle is repeated until a certain termination criterion is meet.

The algorithm proposed is based on the general scheme of an evolutionary algorithm [4], and its pseudocode is showed below.

MS Algorithm with p Parents and c Children.

```
INITIALIZE parents
EVALUATE parents
REPEAT
  FOR i := 1 TO p
    FOR j := 1 TO c
      child[j] = mutate(parent[i])
      evaluate(child[j])
    END FOR
  END FOR
  parents = survivor selection
UNTIL termination criteria is meet
```

Contextualizing the MS algorithm for MBC computation, we have to address the next points:

- The representation and the search space used by the proposed algorithm (described in **subsection 3.2**).
- The *survivor selection* methods used (described in **subsection 3.3**).
- The *children-points* generated through Neighborhood Functions (detailed in **subsection 3.4**).
- The Evaluation function used to measure the quality of the potential solutions (described in **subsection 3.5**).

3.2 Representation and Search Space

The representation used is based on the FNS and the total search space is $r!$ where r is the length of the BC. This representation provides a lower bound denoted by φ and a upper bound denoted by ψ. This bounds are defined in the **Equations** 1 and 2, respectively.

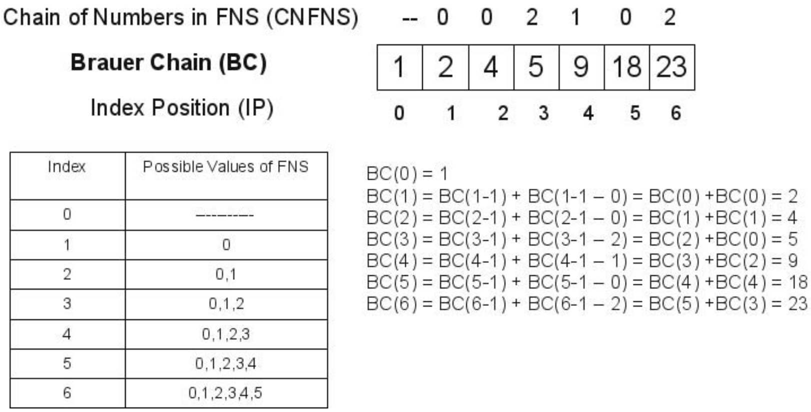
$$\varphi = \log n \tag{1}$$

$$\psi = 2 \cdot \log_2 n \tag{2}$$

The FNS was proposed by Charles-Ange Laisant in [9]. We select it as the representation system because it allows to map a factorial space inside a sequence of digits and also enables to apply some operations like mutations or reproductions without any need to do complex tasks.

In this system, we can describe a BC C with a Chain of Numbers in FNS (CNFNS) by taking a value from the set $\{0, 1, ..., i-1\}$ for each node of C with an index position i greater than 0 such that applying the **Equation** 3 we can rebuild the original BC. To clear the notion of how the FNS is used, the **Figure** 2 show how to represent a BC for $n = 23$ with a CNFNS.

$$BC(i) = \begin{cases} BC(i-1) + BC(i-1-CNFNS(i)) & \textbf{if } i > 0 \\ 1 & \textbf{if } i = 0 \end{cases} \tag{3}$$



Chain of Numbers in FNS (CNFNS)	--	0	0	2	1	0	2
Brauer Chain (BC)	1	2	4	5	9	18	23
Index Position (IP)	0	1	2	3	4	5	6

Index	Possible Values of FNS
0	---------
1	0
2	0,1
3	0,1,2
4	0,1,2,3
5	0,1,2,3,4
6	0,1,2,3,4,5

BC(0) = 1
BC(1) = BC(1-1) + BC(1-1 – 0) = BC(0) +BC(0) = 2
BC(2) = BC(2-1) + BC(2-1 – 0) = BC(1) +BC(1) = 4
BC(3) = BC(3-1) + BC(3-1 – 2) = BC(2) +BC(0) = 5
BC(4) = BC(4-1) + BC(4-1 – 1) = BC(3) +BC(2) = 9
BC(5) = BC(5-1) + BC(5-1 – 0) = BC(4) +BC(4) = 18
BC(6) = BC(6-1) + BC(6-1 – 2) = BC(5) +BC(3) = 23

Fig. 2. How to represent a BC for $n = 23$ with a CNFNS

3.3 Survivor Selection

Eiben (2003) says: "The survivor selection mechanism is responsible for managing the process whereby the working memory of the Genetic Algorithm (GA) is reduced from a set of υ parents and λ offspring to produce the set of υ individuals for the next generation" [4].

We use two types of survivor selection: the first is called “elitist” and takes the best points from the set of *parent-points* and the *children-points*, the second is called “non elitist” and it only takes the best points from the set of the *children-points*.

3.4 Neighborhood Function

The *children-points* are created through a small perturbation of previous solutions (*parent-points*) who are called neighbors. For this process, two neighborhood functions (NF) are proposed: $\mathcal{N}_1(s)$ and $\mathcal{N}_2(s)$, where s is a BC in its CNFNS representation.

- $\mathcal{N}_1(s)$. Select a random index position i from s, and pickup another FNS value different from the original.
- $\mathcal{N}_2(s)$. Select a random index position i from s, and pickup a FNS value different from the original in i. Then select another different random position j from s, and pickup a FNS value different from the original in j.

NFs allows the construction of new possible solutions for BCs with length r by modifying a random point i of the chain such that $(2 \leq i \leq r)$. The process to select a random point is: first calculate a random value x with $0 \leq x \leq \tau$ (**Equation 4**), second use one of the two distribution functions (DF) in **Equation 5** to calculate the i index position.

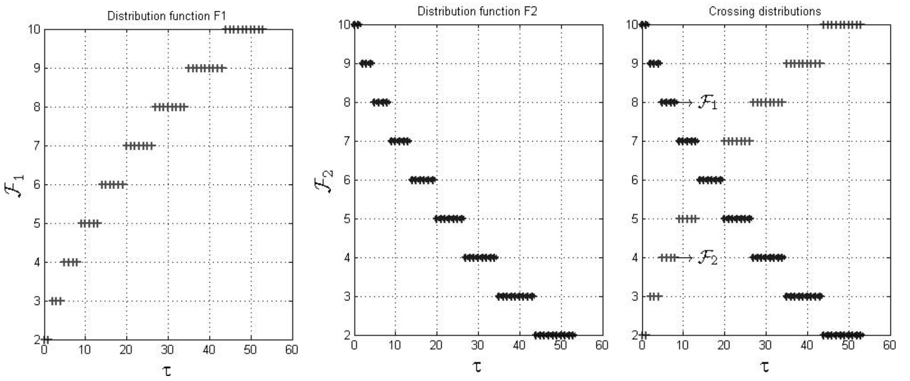
$$\tau = \sum_{i=2}^{r-1} i = \frac{(r-1) \times r}{2} - 1 \tag{4}$$

$$i = \begin{cases} \mathcal{F}_1 = \left\lfloor \frac{1+\sqrt{1+8(x+1)}}{2} \right\rfloor \\ \mathcal{F}_2 = r - \left\lfloor \frac{1+\sqrt{1+8(x+1)}}{2} \right\rfloor \end{cases} \tag{5}$$

Holland (1992) says “If successive populations are produced by mutation only (without reproduction), the results is a random sequence of structures . . . ” [8].

Well, we deal with this attribute of randomness with the use of two distribution functions (DFs), which let us focus in the operations in the chain to get more exploration or exploitation. The DFs $\mathcal{F}_1$ and $\mathcal{F}_2$ are used to determine which parts of the CNFNS sequence is going to be changed with more frequency, it is because changing an i position value closest to the start of the CNFNS chain, the BC in it position r will change drastically, in other words changes are more exploratory. By other side, if the i position value is closest to the value of r, changes does not have a significant effect, the behavior will do more exploitation. We are going to use the NFs and the DFs according with some probabilities that we define later.

The **Figure 3** shows how works the distribution functions, where the *x-axis* are the possible x numbers for a BC with $r = 10$ and the *y-axis* are the corresponding i position of the BC in its FNS representation.



(a) Distribution F_1 (b) Distribution F_2 (c) Distribution F_1 and F_2

Fig. 3. Index positions obtained by using the DFs F_1 and F_2 with all possible values of x for a CNFNS of length 10

3.5 Evaluation Function

The evaluation function Ψ used in the MS algorithm is shown in **Equation 6**.

$$\Psi = r \mid n' - n \mid + r \tag{6}$$

In **Equation 6** r represents the size of the BC that is evaluated, n' is the value of the evaluated chain in its r position and n is the searched value. So, solutions whose n' is near to n, have an evaluation determined by its length. On the other hand solutions whose distance between n' and the searched value is far away, have an evaluation determined by its distance multiplied and plus by its chain length.

3.6 Experimental Design and Fine-Tuning Process

In the proposed approach we explained how the MS algorithm works, but we do not define the values of probabilities of the parameters to use NFs and DFs. With the purpose of having good results for MS, it is necessary to establish some rules about how to mix them. The probability of use $\mathcal{N}_1$ will be p_1 and the probability of use $\mathcal{N}_2$ will be p_2. The possible values for p_1 and p_2 were set up to $0.1, 0.2, 0.3, \ldots, 1.0$ according to the solution of a Diophantine Equation with two variables (**Equation 7**), which results in a test set with 11 different combinations (**Table 2**) of probabilities.

$$p_1 + p_2 = 1.0 \tag{7}$$

The probabilities of use $\mathcal{F}_1$ could be 0%, 25%, 50%, 75% and 100% and the probability of use $\mathcal{F}_2$ is 100% minus the probability of use $\mathcal{F}_1$.

Enumerating the parameters for the algorithm proposed, we have:

- p (*parents-points*) represents the initial *parents-points* generation.
- c (*children-points*) represents the initial *children-points* generation.
- $\mathcal{I}$ (*iterations*) indicates the total number of *iterations* and determines the life cycle of the algorithm..
- $\mathcal{E}$ (*elitist*) indicates the way in which to apply the survivor selection.
- $\mathcal{F}_1$ and $\mathcal{F}_2$ indicates the probability of use the DFs $\mathcal{V}_1$ and $\mathcal{V}_2$ inside the NFs.

Now, the question is, how we are going to set the parameters to get the best performance?. For this question, there are many possible answers like: to search in the literature and obtain the values, make another algorithm that fine-tune the MS or use a Covering Array (CA). Escogido (2008) defines a CA as a bidimensional array of size $N \times k$ where every $N \times t$ subarray contains all ordered subsets from v symbols of size t at least once. The value t is called the strength of the CA, the value k is the number of columns or parameters and v is called the alphabet. The CA is represented as $CA(N; t, k, v)$ [5].

The methodology followed to tune the values of the parameters of the algorithm is based on the study of the effect over the quality of the solution by the interaction between parameters. The tuning process was done using a $CA(25; 2, 6, 3^3 5^2 2^1)$[1] to adjust the parameters of the MS. As already defined, there were $k = 6$ parameters subject to be optimized, and for each parameter, we defined a set of three values ($v = 3$). The interaction between parameters that was set to 2 ($t = 2$), i.e., all the possible combinations between pairs of parameters were analyzed before deciding the best configuration. The $CA(25; 2, 6, 3)$ consists of 25 rows. Each row corresponds to a combination of values for the parameters. Together, all the rows contain all the interactions between pairs of parameter values, used during the experiment.

Also, we tried all the probability combinations of using NFs (the possible solutions of the Diophantine Equation 2) with every row of the CA to get a wide spectrum of how the MS algorithm works.

The **Equation 8** represents the grand total of the experiments that we ran during the fine-tuning process, where $\mathcal{CA}$ represents the number of rows of the CA used (**Table 1a**), $\mathcal{D}$ is the number of possible probability combinations of NFs (Diophantine **Equation 7**) and $\mathcal{B}$ is the number of time that each $\mathcal{CA} \times \mathcal{D}$ experiment was done. For the last parameters of fine-tuning process we set $n = 14143037$ (is difficult to obtain the minimal addition chain of this value as stated in [3]) and to get results with statistical significance we set $\mathcal{B} = 31$.

$$\mathcal{T} = \mathcal{CA} \times \mathcal{D} \times \mathcal{B} \tag{8}$$

Since $\mathcal{CA} = 25$, $\mathcal{D} = 11$, $\mathcal{B} = 31$, then the total number of experiments is $25 \times 11 \times 31 = 8525$. Finally, in this experiment we obtained as the best setting of properties the row of the CA $\boxed{1}\boxed{2}\boxed{2}\boxed{0}\boxed{4}\boxed{2}$ with the solutions of the Diophantine

[1] http://www.tamps.cinvestav.mx/˜jtj

Table 1. CA Values used for the fine tuning process

(a) CA values

Ind	m	n	$\mathcal{I}$	$\mathcal{E}$	$\mathcal{V}_1$	$\mathcal{V}_2$
1	0	1	0	1	1	1
2	0	0	1	0	3	3
3	1	0	1	1	2	3
4	0	1	1	0	2	2
5	1	0	2	0	1	3
6	1	1	1	1	4	1
7	0	0	1	1	2	1
8	1	2	0	1	0	2
9	1	0	1	0	1	4
10	0	0	0	0	4	0
11	1	2	0	1	2	0
12	2	0	0	1	3	0
13	1	2	1	0	0	3
14	2	2	1	0	1	2
15	0	0	2	0	0	0
16	0	2	0	1	2	4
17	0	2	2	0	3	1
18	1	1	2	0	3	4
19	2	0	2	1	2	3
20	2	1	2	1	4	4
21	1	1	0	0	4	3
22	0	1	2	1	0	4
23	2	2	2	0	0	1
24	0	1	1	0	1	0
25	1	2	2	0	4	2

(b) Values for the parameters of the algorithm according the CA values

Values	**0**	**1**	**2**	**3**	**4**
m	$\log_2 \alpha$	$2 \cdot \log_2 \alpha$	$3 \cdot \log_2 \alpha$	-	-
n	$3 \cdot \log_2 \alpha$	$5 \cdot \log_2 \alpha$	$7 \cdot \log_2 \alpha$	-	-
$\mathcal{I}$	$N \times 1000$	$N \times 2000$	$N \times 10000$	-	-
$\mathcal{E}$	non-elitist	elitist	-	-	-
$\mathcal{V}_1$	$0, 1$	$\frac{1}{4}, \frac{3}{4}$	$\frac{2}{4}, \frac{2}{4}$	$\frac{1}{4}, \frac{3}{4}$	$1, 0$
$\mathcal{V}_2$	$0, 1$	$\frac{1}{4}, \frac{3}{4}$	$\frac{2}{4}, \frac{2}{4}$	$\frac{1}{4}, \frac{3}{4}$	$1, 0$

Equation $\boxed{.8}\boxed{.2}$. We also observe that column m with the value 2 of CA, produce better results than others, therefore it was decided to modify the value of the CA line from $\boxed{1}\boxed{2}\boxed{2}\boxed{0}\boxed{4}\boxed{2}$ to $\boxed{2}\boxed{2}\boxed{2}\boxed{0}\boxed{4}\boxed{2}$ and after try the hypothesis, we confirm that it was right and this last configuration improve the results.

3.7 Implementation Note

The proposed MS algorithm was coded using C language and compiled with GCC 4.3.5 with any optimization parameter. The algorithm has been run on a cluster with 4 processor six-core AMD® 8435 (2.6 Ghz), 32 GB RAM, and Operating System Red Hat Linux Enterprise 4.

4 Results

For the experimentation we used the best parameter values found by the fine-tuning process described in the **Section 3.6**, according to it, the better CA row

Table 2. Diophantine Equation values

$NF\backslash number$	1	2	3	4	5	6	7	8	9	10	11
p_1	0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
p_2	1.0	.9	.8	.7	.6	.5	.4	.3	.2	.1	0

was $\boxed{2}\boxed{2}\boxed{2}\boxed{0}\boxed{4}\boxed{2}$ and the best solution of the Diophantine Equation was $\boxed{.8}\boxed{.2}$. Also, each experiment was tested 31 times for the different values of n.

The results generated are shown in **Table 3**, where we can see the set of n's tried, its minimal length, the number of hits obtained (the number of times where a MBC was found) and the following statistics to get a hit: minimal iterations, average iterations, minimum time needed (in seconds) and average time (in seconds). **Table 4** presents some MBC found by the proposed MS algorithm. **Figure 4** represent the maximum and minimum values of length found for the set of n's values in our experiment compared with the optimal values presented in **Table 3**. The hits, times and iterations here showed are acceptable for the length of the set of n's used comparing our results with the follow approaches:

Table 3. Summary of results for the computational experiment

Id	n	**minimal length**	**hits**	**minimal iterations**	**average iterations**	**minimal time (s)**	**average time (s)**
1	7	**4**	31	0	66987.129	0.935	2.873
2	11	**5**	21	0	90510.162	3.246	6.060
3	19	**6**	20	0	123330.073	4.559	9.629
4	23	**6**	31	128791	129386.209	4.350	8.415
5	29	**7**	27	5433	111918.388	4.428	7.470
6	47	**8**	10	1040	70657.857	4.376	9.477
7	55	**8**	31	159333	159666.516	3.724	8.927
8	71	**9**	8	1453	88496.750	12.118	13.408
9	127	**10**	4	4403	48683.111	8.180	12.282
10	191	**11**	2	17849	34664.750	8.678	14.816
11	250	**10**	31	3976	189230.887	7.494	14.089
12	379	**12**	29	38403	217671.135	6.459	14.057
13	607	**13**	21	19293	195292.047	12.701	21.635
14	1087	**14**	8	31665	204811.705	11.733	26.326
15	1903	**15**	26	39549	270447.679	12.513	18.538
16	6271	**17**	9	73566	259490.947	12.607	23.032
17	11231	**18**	20	33068	276716.853	19.038	30.650
18	18287	**19**	4	114936	280630.625	20.949	28.136
19	34303	**21**	1	447889	447889.000	29.623	29.623
20	65131	**21**	3	79028	377439.000	29.481	32.396
21	685951	**25**	2	489636	551264.800	23.089	37.696
22	1176431	**27**	7	201197	548469.142	41.996	46.717
23	4169527	**28**	1	630746	630746.000	33.291	33.291
24	7624319	**29**	1	187047	498590.333	27.892	42.237
25	14143037	**30**	1	592150	592150.000	64.844	64.844

- Some of the MACs found by Cortés et al (2005) [3] are for n equal to 34303, 65131, 110599 and 685951.
- Among the results of Nedjah and Mourelle (2003) are the MAC for n equal to 23, 55 and 95.
- Thurber (1999) finds the MACs for n equal to 127, 191 and 607 in 0.82, 7.09 and 130 seconds respectively.
- Bleichenbacher and Flammenkamp (1997) compute a set of MACs among which are: 1, 3, 7, 29, 127, 1903 and 65131.

Table 4. Some MBCs found

n	**BC of minimal length**	**optimal length**
4169527	1→2→3→4→7→14→28→56→112→224→448 →896→1792→1806→3612→7224→14448→14476 →28952→28955→57910→115820→231640→463280 →521190→1042380→ 2084760→4169520→4169527	**28**
7624319	1→2→3→6→9→11→20→29→58→116→232 →464→928→1856→3712→7424→7444→14888 →29776→29782→59564→119128→238256→476512 →953024→1906048→1906077→3812154→7624308 →7624319	**29**
14143037	1→2→3→5→10→20→40→80→83→123→246 →492→575→1150→2300→4600→9200→18400 →36800→73600→147200→147323→294646→589292 →589293→1178586→1767879→3535758→7071516 →7071521→14143037	**30**

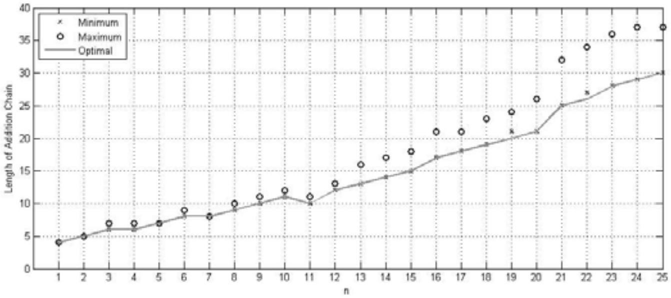


Fig. 4. Comparison of minimum and maximum length (*y-axis*) of BC obtained with the proposed approach versus optimal values for different $n's$ (*x-axis*)

5 Conclusions

The quality of our experimental results demonstrated the strength of each part of the proposed approach: the representation based on the FNS help us to do the operation of mutation without repairing each solution, also this representation could be used by another metaheuristics algorithms (like genetic algorithms) because it provides flexibility to accept other kind of operations like recombination and inversion; the use of no fixed parameters for the use NFs and DFs, enabled the experimentation with a wide range of the possible behaviors of the algorithm, but increased the number of parameters to be adjusted; in this sense, the fine-tuning process, using a DE and a CA give us the possibility to uncover excellent parameter values and obtained the best performance of the MS algorithm without the need to make a lot of experiments.

The results obtained from the proposed approach provided the solution of the minimum BC problem even to particular benchmarks considered difficult. In this sense, our algorithm finds a MBC in less than $3000 \cdot \log_2 n$ iterations and 1.5 minutes for the hardest n tried.

We suggest to follow a fine-tuning methodology based in the use of Covering Arrays and Diophantine Equations in order to get really good values for the parameters of an algorithm avoiding a long and complex process of parameter optimization.

There is still a lot of work to get an efficient and optimal algorithm to solve the problem of get MBCs, but the proposed approach openned another way to face it by mixing a genetic algorithm using FNS and no fixed GA operators.

Acknowledgments. This research was partially funded by the following projects: CONACyT 58554 - Cálculo de Covering Arrays, 51623 - Fondo Mixto CONACyT y Gobierno del Estado de Tamaulipas.

References

1. Bleichenbacher, D., Flammenkamp, A.: Algorithm for computing shortest additions chains. Tech. rep., Bell Labs (1997), wwwhomes.uni-bielefeld.de/achim/ac.ps.gz
2. Brauer, A.: On addition chains. Jahresbericht der deutschen Mathematiker-Vereinigung 47, 41 (1937)
3. Cruz-Cortés, N., Rodríguez-Henríquez, F., Juárez-Morales, R., Coello Coello, C.A.: Finding optimal addition chains using a genetic algorithm approach. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-m., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS (LNAI), vol. 3801, pp. 208–215. Springer, Heidelberg (2005)
4. Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
5. Lopez-Escogido, D., Torres-Jimenez, J., Rodriguez-Tello, E., Rangel-Valdez, N.: Strength two covering arrays construction using a SAT representation. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS (LNAI), vol. 5317, pp. 44–53. Springer, Heidelberg (2008)
6. Gelgi, F., Onus, M.: Heuristics for minimum brauer chain problem, vol. 47, pp. 47–54. Springer, Heidelberg (2006)
7. Guy, R.K.: Unsolved problems in mathematics in the field of number theory, 3rd edn. Springer, Heidelberg (2004)
8. Holland, J.: Adaptation in natural and artificial systems. MIT Press (1992)

9. Laisant, C.: Sur la numération factorielle, application aux permutations (in French). Bulletin de la Société Mathématique de France 16 (1888)
10. Michalewicz, Z.: Genetic algorithms + data structures = evolution program, 3rd edn. Springer, Heidelberg (1996)
11. Nedjah, N., Mourelle, L.M.: Efficient parallel modular exponentiation algorithm, pp. 405–414. Springer, Heidelberg (2002)
12. Nedjah, N., Mourelle, L.M.: Efficient pre-processing for large window-based modular exponentiation using genetic algorithms. In: Chung, P.W.H., Hinde, C.J., Ali, M. (eds.) IEA/AIE 2003. LNCS, vol. 2718, pp. 165–194. Springer, Heidelberg (2003)
13. Thurber, E.: Efficient generation of minimal length addition chains. Journal on Computing 28(4) (1999)

Hyperheuristic for the Parameter Tuning of a Bio-Inspired Algorithm of Query Routing in P2P Networks

Paula Hernández[1], Claudia Gómez[1], Laura Cruz[1], Alberto Ochoa[2],
Norberto Castillo[1] and Gilberto Rivera[1]

[1] División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de Ciudad Madero. Juventino Rosas y Jesús Urueta s/n,
Col. Los mangos, C.P. 89440, Cd. Madero, Tamaulipas, México
{paulahdz314,cggs71}@hotmail.com, lcruzreyes@prodigy.net.mx,
{norberto_castillo15,grivera984}@hotmail.com
[2] Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez. Henry
Dunant 4016, Zona Pronaf, C.P. 32310, Cd. Juárez, Chihuahua, México
doctor_albertoochoa@hotmail.com

Abstract. The computational optimization field defines the parameter tuning problem as the correct selection of the parameter values in order to stabilize the behavior of the algorithms. This paper deals the parameters tuning in dynamic and large-scale conditions for an algorithm that solves the *Semantic Query Routing Problem* (SQRP) in *peer-to-peer* networks. In order to solve SQRP, the HH_AdaNAS algorithm is proposed, which is an ant colony algorithm that deals synchronously with two processes. The first process consists in generating a SQRP solution. The second one, on the other hand, has the goal to adjust the *Time To Live* parameter of each ant, through a hyperheuristic. HH_AdaNAS performs adaptive control through the hyperheuristic considering SQRP local conditions. The experimental results show that HH_AdaNAS, incorporating the techniques of parameters tuning with hyperheuristics, increases its performance by 2.42% compared with the algorithms to solve SQRP found in literature.

Keywords: Parameter Tuning, Hyperheuristic, SQRP.

1 Introduction

Currently, the use of evolutionary computation has become very popular as a tool to provide solutions to various real-world problems. However, different tools proposed in the evolutionary field require careful adjustment of its parameters, which is usually done empirically, and is also different for each problem to be solved. It should be mentioned that specialized adjustment leads to an increase in the development cost.

The parameter tuning problem has received a lot of attention, because the efficiency of the algorithms is significantly affected by the assigned value to its parameters.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 119–130, 2011.

There are few papers which deal the parameter tuning in dynamic and large-scale conditions, such as the *Semantic Query Routing Problem* (SQRP) in *peer-to-peer* (P2P).

SQRP is a complex problem that has characteristics that are challenging for search algorithms. Due to its difficulty this problem has been partially developed under different perspectives [1][2][3]. The works mentioned above, use as solution technique ant colony algorithms. In these algorithms the TTL parameter, which indicates the maximum allowed time for each query in the network, begins with a static value and is decreased gradually by a fixed rule. More recent works such as Rivera [4] and Gomez [5] have focused on using adaptive techniques for adjusting this parameter considered significant [6]. In this work, when the TTL runs out, the algorithm uses an adaptive strategy to decide whether or not to extend the time to live.

In this paper, the main motivation was to create an algorithm called HH_AdaNAS with adaptive techniques through hyperheuristic strategies. The adaptation is performed throughout the search process. This feature makes the difference with such works, because the hyperheuristic defines itself and during its execution, the appropriate TTL values.

So HH_AdaNAS is an ant colony algorithm that deals synchronously with two processes. The first process consists in generating a SQRP solution. The second one, on the other hand, has the goal to adjust the *Time To Live* parameter of each ant, through of the hyperheuristic proposed.

Moreover, after a literature search, we found that SQRP has not been dealt with hyperheuristic techniques, these techniques have been used in other application domains, some of them are: Packing [7] and Vehicle Routing Problem [8]. It should be mentioned that few researchers have tackled the adaptation of parameters in hyperheuristics [9][10].

2 Background

This section describes the information related to research. First hyperheuristic term is defined, after the parameter tuning, the semantic query routing and P2P networks are described.

2.1 Hyperheuristic

A hyperheuristic is a high-level algorithm that acts as a planner on a set of heuristics that makes the programming in a deterministic or nondeterministic form [11]. The most appropriate heuristic is determined and is automatically applied by the hyperheuristic technique at each step to solve a given problem [12].

2.2 Parameter Tuning

Each one of the combinations of parameter values is called *parametric configuration*, and the problem of selecting appropriate values for the parameters to regulate the behavior of algorithms is called parameter tuning [13][14].

The classification proposed by Michalewicz & Fogel [15] divides the parameter tuning in two stages depending on what part of the experiment is applied. If applied before the execution of the experiment it is called *parameter control.*

The parameter control is divided into deterministic, adaptive and self-adaptive control. *Adaptive control*, which is performed in this work, is done when there is some form of feedback from the past that determines a change in direction and magnitude of the parameter.

2.3 Routing of Semantic Consultation and Peer to Peer Nets

The problem of searching for textual information through keywords on Internet is known as *Semantic Query Routing* (SQRP). Its objective is to determine the shortest path from a node that issues a query to the location of the nodes that can answer it appropriately providing the required information. Complex systems such as SQRP involve elements such as the environment (topology), entities that interact in the system (nodes, repositories and queries) and an objective function (minimizing steps and maximizing results) [2][16]. This problem has been taking a great relevance with the growth of the peer-to-peer communities.

Peer to peer systems are defined as distributed systems consisting of interconnected nodes that have equal role and responsibility. These systems are characterized by decentralized control, scalability and extreme dynamism of their operating environment [17][18]. Some examples include academic P2P networks, such as LionShare [19] and military networks, such as DARPA [20].

3 Description of HH_AdaNAS

This section presents the architecture of the system, data structures, the description of the proposed algorithm HH_AdaNAS and the description of hyperheuristic HH_TTL implemented.

3.1 Architecture of HH_AdaNAS

HH_AdaNAS is adaptive metaheuristic algorithm, based on AdaNAS [4], but incorporates a hyperheuristic called HH_TTL; it adapts the parameter of time to live during the execution of the algorithm. HH_AdaNAS uses as solution algorithm an Ant Colony.

This algorithm has two objectives: it seeks to maximize the number of resources found by the ants and to minimize the number of steps that the ants take it. The general architecture of the multi-agents system HH_AdaNAS is shown in Figure 1, and comprises two main elements:

1. Environment *E*, which is a static P2P complex network.
2. Agents $\{w, x, y, z, x_{hh}, z_{hh}\}$. HH_AdaNAS has six types of agents, each of which have a specific role. They are represented as ants of the algorithm HH_AdaNAS proposed, these ants modify the environment and the hyperheuristic ants x_{hh} and z_{hh} adapts the TTL parameter. The function of each agent is described in Section 3.2.

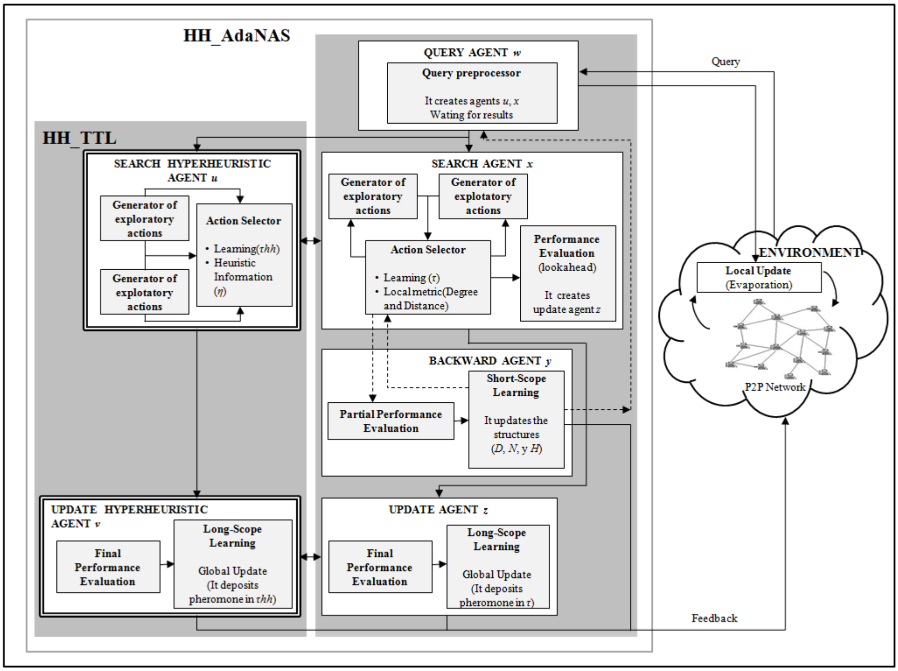

HH_AdaNAS
QUERY AGENT *w*
Query preprocessor
It creates agents *u*, *x*
Wating for results
Query
HH_TTL
SEARCH HYPERHEURISTIC AGENT *u*
Generator of exploratory actions
Action Selector
• Learning(*τhh*)
• Heuristic Information (*η*)
Generator of explotatory actions
SEARCH AGENT *x*
Generator of exploratory actions
Generator of explotatory actions
Action Selector
• Learning (*τ*)
• Local metric(Degree and Distance)
Performance Evaluation (lookahead)
It creates update agent *z*
ENVIRONMENT
Local Update (Evaporation)
P2P Network
BACKWARD AGENT *y*
Short-Scope Learning
It updates the structures (*D*, *N*, y *H*)
Partial Performance Evaluation
UPDATE HYPERHEURISTIC AGENT *v*
Final Performance Evaluation
Long-Scope Learning
Global Update (It deposits pheromone in *τhh*)
UPDATE AGENT *z*
Final Performance Evaluation
Long-Scope Learning
Global Update (It deposits pheromone in *τ*)
Feedback

Fig. 1. General Architecture of HH_AdaNAS

3.2 Data Structures of HH_AdaNAS

The proposed algorithm HH_AdaNAS consists of six data structures, in which are stored heuristic information or gained experience in the past. The relationship of these structures is shown in Figure 2.

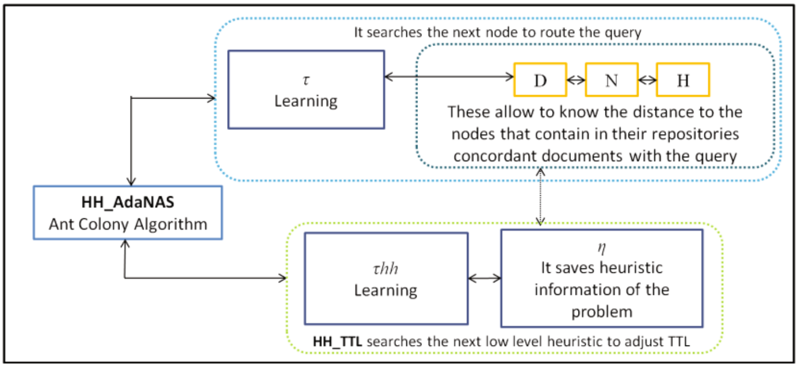
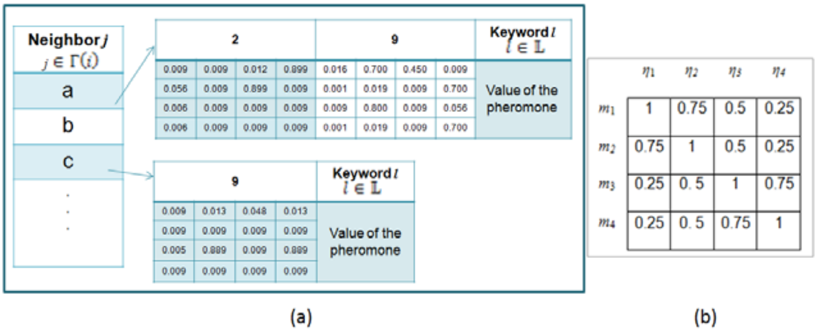

It searches the next node to route the query
τ
Learning
D
N
H
These allow to know the distance to the nodes that contain in their repositories concordant documents with the query
HH_AdaNAS
Ant Colony Algorithm
τhh
Learning
η
It saves heuristic information of the problem
HH_TTL searches the next low level heuristic to adjust TTL

Fig. 2. Data structures of HH_AdaNAS

When HH_AdaNAS searches for the next node, in the routing process of the query, is based on the pheromone table τ and tables D, N y H [21]. Also, when HH_TTL chooses the following low level heuristic through Equation 1 is based on the following tables:

1. The *pheromone* table τhh is divided into n two-dimensional tables, corresponding one τhh_i for each node i in the network. Each $\tau hh_{i,j,l}$ in turn contains a two-dimensional table $|m| \times |n|$, where m is the number of visibility states of the problem and n is the total number of heuristics; an example of this can be seen in Figure 3a.
2. The table of *visibility states* η is of size $|m|\text{x}|n|$ and is shown in the Figure 3b. The values of the table η were assigned according to knowledge of the problem and they are static.



2				9				Keyword l $l \in \mathbb{L}$
0.009	0.009	0.012	0.899	0.016	0.700	0.450	0.009	Value of the pheromone
0.056	0.009	0.899	0.009	0.001	0.019	0.009	0.700	
0.006	0.009	0.009	0.009	0.009	0.800	0.009	0.056	
0.006	0.009	0.009	0.009	0.001	0.019	0.009	0.700	

9				Keyword l $l \in \mathbb{L}$
0.009	0.013	0.048	0.013	Value of the pheromone
0.009	0.009	0.009	0.009	
0.005	0.889	0.009	0.889	
0.009	0.009	0.009	0.009	

	η_1	η_2	η_3	η_4
m_1	1	0.75	0.5	0.25
m_2	0.75	1	0.5	0.25
m_3	0.25	0. 5	1	0.75
m_4	0.25	0. 5	0.75	1

Fig. 3. Data structures of the hyperheuristic HH_TTL. a) Pheromone table τhh and b) Table of the visibility states η.

3.3 Algorithmic Description of HH_AdaNAS

In parallel all the queries in the HH_AdaNAS use *Query Ants w*. Each ant *w* generates a *Forward Ant x* (It generates a solution for SQRP) and *Hyperheuristic Forward Ant u* (It adjusts adaptively the TTL parameter), besides, this ant updates the pheromone tables τ and τhh though the evaporation.

Algorithm 2 shows the routing process, which is performed by the *Forward Ant x* and *Hyperheuristic Forward Ant u*, these ants work synchronously (see Figure 1). All the ants work in parallel.

In the beginning, ant *u* has a time to live of TTL_{inic}. The operation of the algorithm can be divided into three phases. In an initial phase (lines 4-8), the ant *x* checks the local repository of the issuing node of the query and, if documents are consistent, creates a *Backward Ant y*, the algorithm followed by the *Backward Ant y* is found in Gomez et al. [21]. The *Backward Ant y* informs to *Query Ant w* the amount of found resources on a node by the *Forward Ant x* and updates the values of some learning structures (*D*, *N* and *H*).

Subsequently, the next phase is the search process (lines 9-22), which is performed until the time to live runs out and are not *R* consistent documents. *R* is the number of documents required by users.

During the search process results are evaluated (lines 10-15) [3], next node is selected (lines 16-18 and 20) [4] and the time to live parameter is adjust by proposed hyperheuristic HH_TTL (lines 19 and 21).

HH_TTL, through *Hyperheuristic Forward Ant u* selects the low level heuristic that best adapts TTL, this by Equation 1 (Line 19). *Sequence_TTL* structure is the sequence of heuristics that make adapting the TTL parameter, this structure is updated in line 21.

In the final phase of the algorithm HH_AdaNAS (lines 23-28) the *Forward Ant x* creates *Update Ant z* and evaluates the solution generated for SQRP, the rule is described in Gomez et al. [21]. Also *Hyperheuristic Forward Ant u* creates *Hyperheuristic Update Ant v* and the last one deposits the pheromone on the path traveled by the ant *u* (line 24), that is, the sequence of low level heuristics selected for the adaptation of TTL. The deposit rule for the table τhh is shown in Equation 6.

Algorithm 2. HH_AdaNAS Algorithm that show the routing process with hyperheuristic

```
1    Processs in parallel for each Forward Ant x (r, l) and each Hyperheuristic
     Forward Ant u (m, n)
2       Initialization: path ← ⟨r⟩, Λ ← {r}, known ← {r}
3       Initialization: TTL = $TTL_{inic}$, sequence_TTL ← ⟨n⟩
4       results ← get local documents(r)
5       If results > 0 then
6          Create Backward Ant y (path, results, l)
7          Activate y
8       End
9       While TTL > 0 and results < R do
10         la_results ← lookahead(r, l, known)
11         If la_results > 0 then
12            Create Backward Ant y (path, results, l)
13            Activate y
14            results ← results + la_results
15         End
16         known ← known ∪ Γ(r)
17         Λ ← Λ ∪ r
18         Apply transition rule: r ← $\ell(x, r, l)$
19         Apply Adaptation_TTL rule: n ← $\phi(u,r,s,l,m)$
20         add_to_path(r)
21         add_to_sequece_TTL(n)
22      End
23      Create Update Ant z (x, path, l)
24      Create Hyperheuristic Update Ant v (u, path, sequence_TTL, l)
25      Activate z
26      Activate v
27      Kill x
28      Kill u
29   End of the Process in parallel
```

3.4 Description of HH_TTL

The hyperheuristic, which adapts the time to live (*Hyperheuristic_Time To Live*, HH_TTL), was designed with online learning [12], and uses an Ant Colony metaheuristic as high level heuristic.

As shown in Figure 4, the low level heuristics are related with SQRP. It also notes that there is a barrier between the hyperheuristic and the set of low level heuristics; this allows the hyperheuristic to be independent of the problem domain. In this context, hyperheuristic would ask how each of the low-level heuristics would work, so it can decides which heuristic to apply at each time to adapt the TTL parameter, according to the current state of the system, in this case, of performance achieved.

The design of the hyperheuristic was done so that while the solution is built for SQRP, low-level heuristics adapt the TTL parameter, this working synchronously.

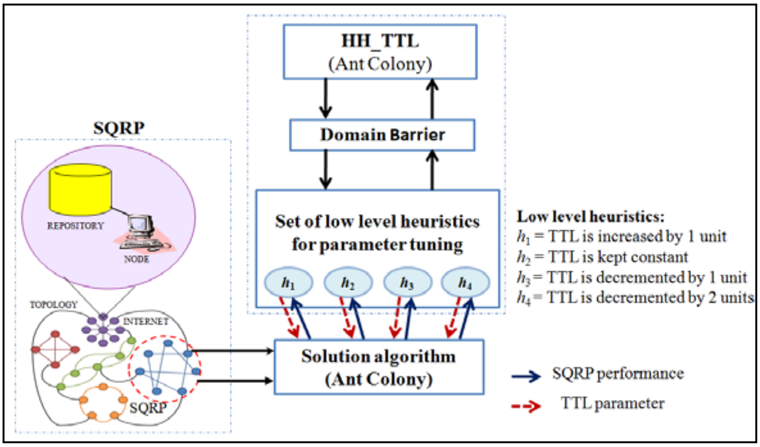


Fig. 4. General diagram of HH_TTL

3.5 Rules of Behavior of HH_TTL

The hyperheuristic HH_TTL has two rules of behavior, which interact with data structures: the selection rule and update rules.

1. Selection Rule of the Heuristics

In this stage the *Hyperheuristic Forward Ant u*, selects the low level heuristic to adapt TTL. This movement is realized following a selection rule that uses local information, which includes heuristic information η and learning (table τhh) to guide the search.

First HH_TTL determines the state m of SQRP, in which is the *Hyperheuristic Forward Ant u*, after that selects the best low level heuristic n that adapts the TTL parameter.

The selection rule for *Hyperheuristic Forward Ant u*, which is consulted trough keyword l, located at node r and it decided to route the query to node s, with the visibility state m is the following:

$$\phi(u,r,s,l,m) = \begin{cases} \arg\max_{n\in \mathrm{H}} \{\delta(r,s,l,m,n)\}, & si\ \varphi \leq q \\ \gamma(u,r,s,l,m), & otherwise \end{cases} \tag{1}$$

where $\phi(u,r,s,l,m)$ is the function that selects the next low level heuristic, φ is a number pseudorandom, q is an algorithm parameter which defines the probability of using the exploitation or exploration technique, φ and q acquires values between zero and one. H is the set of low level heuristics of the visibility state *m* and the Equation 2 shows the exploration technique,

$$\delta(r,s,l,m,n) = \left(\eta_{m,n}\right)^{\beta_1}\left(\tau hh_{r,s,l,m,n}\right)^{\beta_2} \tag{2}$$

where β_1 is a parameter that intensify the contribution of the visibility $(\eta_{m,n})$ and β_2 intensify the contribution of the pheromone $(\tau hh_{r,s,l,m,n})$. The table η has heuristic information of the problem and the pheromone table τhh saves the gained experience in the past.

In the Equation 1, γ is exploration technique, which selects the next low level heuristic. This technique is expressed as:

$$\gamma(u,r,s,l,m) = f\left(\{p_{u,r,s,l,m,n} | n \in H\}\right) \tag{3}$$

where $f\left(\{p_{u,r,s,l,m,n} | n \in H\}\right)$ is the roulette-wheel random selection function which selects low level heuristic *n* depending on its $p_{u,r,s,l,m,n}$, which indicates the probability that the *Hyperheuristic Forward Ant u*, which is the visibility state *m*, selects the heuristic *n* as the following in the adaptation of TTL. It can define $p_{u,r,s,l,m,n}$ as:

$$p_{u,r,s,l,m,n} = \frac{\delta(r,s,l,m,n)}{\sum_{n\in \mathrm{H}} \delta(r,s,l,m,n)} \tag{4}$$

2. Update Rules of the Hyperheuristic

The proposed hyperheuristic HH_TTL applies deposit and evaporation rules on its pheromone table τhh.

Evaporation Rule of the Pheromone

When choosing a low level heuristic, the proposed hyperheuristic algorithm implements a local update on the table τhh, in each unit of time (typically 100 ms), which is the following:

$$\begin{gathered} \tau hh_{r,s,l,m,n} = (1-\rho)\tau hh_{r,s,l,m,n} + \rho\tau_0 \\ \forall (r,s,l,m,n) \in \{r\} \times \Gamma(i) \times \mathbb{L} \times \mathbb{S} \times \mathrm{H}, \end{gathered} \tag{5}$$

where *r* is the current node, *s* is the selected node to route the query by the keyword *l*, *m* is the current visibility state, *n* is the select heuristic, ρ is the evaporation rate of pheromone (number between zero and one) and τ_0 is the initial value of pheromone. $\mathbb{L}$ is the dictionary for the queries, $\mathbb{S}$ is the set of the visibility state, H is the set of low level heuristics and $\{r\} \times \Gamma(i) \times \mathbb{L}$ is the Cartesian product between sets $\{r\}$, $\Gamma(i)$, $\mathbb{L}$, $\mathbb{S}$ and H .

Deposit Rule of the Pheromone

Once each *Hyperheuristic Forward Ant u* has generated a solution, it is evaluated and an amount of pheromone is deposited, that is based on the quality of its solution. This process is realized by a *Hyperheuristic Update Ant v*.

When the *Hyperheuristic Update Ant v* is created runs in reverse the route generated by the *Hyperheuristic Forward Ant u,* whenever it reaches a different heuristic modifies the pheromone trail according to the formula:

$$\tau hh_{r,s,l,m,n} = \tau hh_{r,s,l,m,n} + \Delta\tau hh_{r,s,l,m,n}(u) \tag{6}$$

In the Equation 6, $\tau hh_{r,s,l,m,n}$ is the preference of selecting the low level heuristic *n*, in the state m, for *Hyperheuristic Forward Ant u* located in the node *r*, which has selected the node s to route the query by *l*. $\Delta\tau hh_{r,s,l,m,n}(u)$ is the amount of pheromone deposited by *Hyperheuristic Update Ant v* and

$$\Delta\tau hh_{r,s,l,m,n}(u) = (w_h)\frac{hits(x,s)}{R} + (1 - w_h)\frac{1}{hops(x,r)} \tag{7}$$

where R is the amount of required resources, w_h is an parameter that represents the goodness of the path and takes a value between zero and one, $hits(x,s)$ is the amount of found resources by the *Forward Ant x* from node s until the end of its route, $hops(x,r)$ is the length of the generated route by *Forward Ant x* from node *r* to the end of its route.

4 Experimental Results

This section presents the performance of the algorithm and is compared with an algorithm of the literature in the area. It also describes the experimental setup and test instances used.

4.1 Experimental Environment

The following configuration corresponds to the experimental conditions that are common to the test described.

Software: Operative system Microsoft Windows 7 Home Premium; Java programming language, Java Platform, JDK 1.6; and integrated development, Eclipse 3.4.

Hardware: Computer equipment with processor Intel (R) Core (TM) i5 CPU M430 2.27 GHz and RAM memory of 4 GB.

Instances: It has 90 different SQRP instances; each of them consists of three files that represent the topology, queries and repositories. The description of the features can be found in Cruz et al. [6].

Initial Configuration of HH_AdaNAS

Table 1 shows the assignment of values for each HH_AdaNAS parameter. The parameter values were based on values suggested of the literature as Dorigo [22], Michlmayr [2], Aguirre [3] and Rivera [4].

Table 1. Values for the parameters of HH_AdaNAS

Parameter	Description	Value
τ_0	Pheromone table initialization	0.009
D_0	Distance table initialization	999
ρ	Local pheromone evaporation factor	0.35
β_1	Intensification of local measurements (degree and distance)	2.0
β_2	Intensification of pheromone trail	1.0
q	Relative importance between exploration and Exploitation	0.65
W_h	Relative importance of the hits and hops in the increment rule	0.5
W_{deg}	Degree's influence in the selection the next node	2.0
W_{dist}	Distance's influence in the selection the next node	1.0
TTL_{inic}	Initial Time To Live of the Forward Ants	10

4.2 Performance Measurement of HH_AdaNAS

In this section we show experimentally that our HH_AdaNAS algorithm outperforms the AdaNAS algorithm. Also HH_AdaNAS outperforms NAS, SemAnt and random walk algorithms, inasmuch as in Gomez et al. [21] and Rivera [4] reported that AdaNAS surpasses the NAS performance. Also Gomez et al. [16] reported that NAS outperforms SemAnt and random walk algorithms [3], so our algorithm is positioned as the best of them.

In this experiment, in the HH_AdaNAS and AdaNAS algorithms, the performance achieved by the *Forward Ant x*, which is the agent performing the query, is measured by the rate of found documents by traversed edge. The larger number of found documents by edge that runs the *Forward Ant x*, the better algorithm's performance will have.

To measure the performance of the entire ant colony, the average performance of 100 queries is calculated. The average performance of the latest 100 ants is called *final efficiency of the algorithm*; this measure was used to compare the HH_AdaNAS algorithm with the AdaNAS algorithm.

Each algorithm was run thirty times per instance with the configuration described in Table 1. Figure 5 shows a comparison chart between the resulting performance of the HH_AdaNAS algorithm and the reference algorithm AdaNAS, for each ninety different test instances. It is observed that the HH_AdaNAS algorithm outperforms AdaNAS algorithm. This is because the HH_AdaNAS algorithm achieved an average performance of 2.34 resources by edge resources, while the average performance reached achieved by AdaNAS algorithm was of 2.28 resources by edge. That is, HH_AdaNAS using hyperheuristic techniques had an improvement of 2.42% in average efficiency over AdaNAS. It is because the hyperheuristic HH_TTL in HH_AdaNAS defines itself and during its execution, the appropriate TTL values; and on the other hand, AdaNAS defines the TTL values in a partial and deterministic way.

Additionally, to validate the performance results of these two algorithms, non- parametric statistical test of Wilcoxon was performed [23]. The results of this test reveals that the performance of the algorithm HH_AdaNAS shows a significant improvement over the algorithm AdaNAS, on the set of the 90 test instances, at a confidence level above 95%.

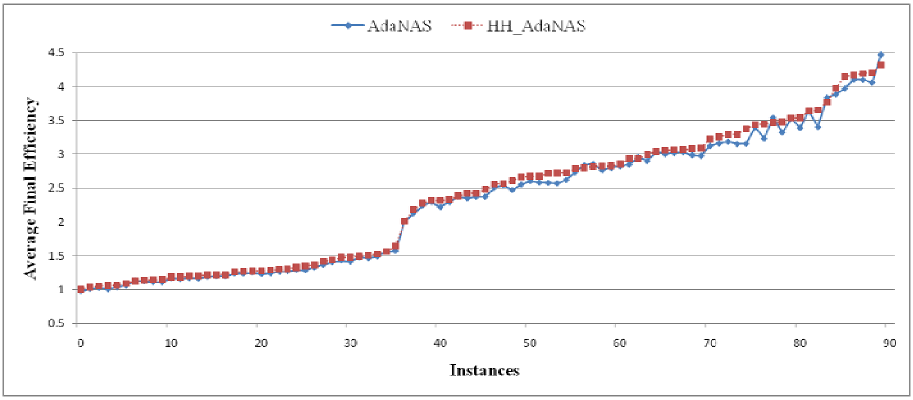


Fig. 5. Comparison of performance between the algorithms HH_AdaNAS and AdaNAS

5 Conclusions

In this work the semantic query routing process was optimized by creating a hyperheuristic algorithm whose main characteristic was its adaptability to the environment. The HH_AdaNAS algorithm was able to integrate the routing process that AdaNAS algorithm performs and the HH_TTL hyperheuristic, which adapts the TTL parameter.

The HH_AdaNAS algorithm has better average performance than his predecessor AdaNAS in 2.42%, taking into account the final efficiency of the algorithms. In the process of adaptation hyperheuristics agents (hyperheuristic ants) do not depend entirely on TTL_{inic} parameter, but is able to determine the necessary time to live while the query is routed to the nodes that satisfy it.

The main difference in the adaptation of the TTL parameter between the algorithms AdaNAS and HH_AdaNAS is that the first one does it in a partial and deterministic form, while the second one does it through the learning acquired during the solution algorithmic process.

References

1. Yang, K., Wu, C., Ho, J.: AntSearch: An ant search algorithm in unstructured peer-to-peer networks. IEICE Transactions on Communications 89(9), 2300–2308 (2006)
2. Michlmayr, E.: Ant Algorithms for Self-Organization in Social Networks. PhD thesis, Women's Postgraduate College for Internet Technologies, WIT (2007)
3. Aguirre, M.: Algoritmo de Búsqueda Semántica para Redes P2P Complejas. Master's thesis, División de Estudio de Posgrado e Investigación (2008)
4. Rivera, G.: Ajuste Adaptativo de un Algoritmo de Enrutamiento de Consultas Semánticas en Redes P2P. Master's thesis, División de Estudio de Posgrado e Investigación, Instituto Tecnológico de Ciudad Madero (2009)
5. Gómez, C.: Afinación Estática Global de Redes Complejas y Control Dinámico Local de la Función de Tiempo de Vida en el Problema de Direccionamiento de Consultas Semánticas. PhD thesis, Instituto Politécnico Nacional, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Unidad Altamira (2009)

6. Cruz, L., Gómez, C., Aguirre, M., Schaeffer, S., Turrubiates, T., Ortega, R., Fraire, H.: NAS algorithm for semantic query routing systems in complex networks. In: DCAI. Advances in Soft Computing, vol. 50, pp. 284–292. Springer, Heidelberg (2008)
7. Garrido, P., Riff, M.-C.: Collaboration Between Hyperheuristics to Solve Strip-Packing Problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 698–707. Springer, Heidelberg (2007)
8. Garrido, P., Castro, C.: Stable Solving of CVRPs Using Hyperheuristics. In: GECCO 2009, Montréal, Québec, Canada, July 8-12 (2009)
9. Han, L., Kendall, G.: Investigation of a Tabu Assisted Hyper-Heuristic Genetic Algorithm. In: Congress on Evolutionary Computation, Canberra, Australia, pp. 2230–2237 (2003)
10. Cowling, P., Kendall, G., Soubeiga, E.: A Hyperheuristic Approach to Scheduling a Sales Summit. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 176–190. Springer, Heidelberg (2001)
11. Özcan, E., Bilgin, B., Korkmaz, E.: A Comprehensive Analysis of Hyper-heuristics. Journal Intelligent Data Analysis. Computer & Communication Sciences 12(1), 3–23 (2008)
12. Burke, E.K., Hyde, M.R., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.R.: Exploring Hyper-Heuristic Methodologies With Genetic Programming. In: Mumford, C.L., Jain, L.C. (eds.) Computational Intelligence. ISRL, vol. 1, pp. 177–201. Springer, Heidelberg (2009)
13. Eiben, A., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3(2), 124–141 (1999)
14. Birattari, M.: The Problem of Tuning Metaheuristics as seen from a machine learning perspective. PhD thesis, Universidad libre de Bruxelles (2004)
15. Michalewicz, Z., Fogel, D.: How to Solve It: Modern Heuristics. segunda edición. Springer, Heidelberg (2004)
16. Gómez, C.G., Cruz, L., Meza, E., Schaeffer, E., Castilla, G.: A Self-Adaptive Ant Colony System for Semantic Query Routing Problem in P2P Networks. Computación y Sistemas 13(4), 433–448 (2010) ISSN 1405-5546
17. Montresor, A., Meling, H., Babaoglu, Ö.: Towards Adaptive, Resilient and Self-organizing Peer-to-Peer Systems. In: Gregori, E., Cherkasova, L., Cugola, G., Panzieri, F., Picco, G.P. (eds.) NETWORKING 2002. LNCS, vol. 2376, pp. 300–305. Springer, Heidelberg (2002)
18. Ardenghi, J., Echaiz, J., Cenci, K., Chuburu, M., Friedrich, G., García, R., Gutierrez, L., De Matteis, L., Caballero, J.P.: Características de Grids vs. Sistemas Peer-to-Peer y su posible Conjunción. In: IX Workshop de Investigadores en Ciencias de la Computación (WICC 2007), pp. 587–590 (2007) ISBN 978-950-763-075-0
19. Halm M., LionShare: Secure P2P Collaboration for Academic Networks. In: EDUCAUSE Annual Conference (2006)
20. Defense Advanced Research Project Agency (2008), `http://www.darpa.mil`
21. Santillán, C.G., Reyes, L.C., Schaeffer, E., Meza, E., Zarate, G.R.: Local Survival Rule for Steer an Adaptive Ant-Colony Algorithm in Complex Systems. In: Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Recognition Based on Biometrics. SCI, vol. 312, pp. 245–265. Springer, Heidelberg (2010)
22. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
23. García, S., Molina, D., Lozano, F., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC 2005 Special Session on Real ParameterOptimization. Journal of Heuristics (2008)

Bio-Inspired Optimization Methods for Minimization of Complex Mathematical Functions

Fevrier Valdez, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology, Tijuana, B.C.
{fevrier,pmelin,ocastillo}@tectijuana.mx

Abstract. This paper describes a hybrid approach for optimization combining Particle Swarm Optimization (PSO) and Genetic Algorithms (GAs) using Fuzzy Logic to integrate the results, the proposed method is called FPSO+FGA. The new hybrid FPSO+FGA approach is compared with the Simulated Annealing (SA), PSO, GA, Pattern Search (PS) methods with a set of benchmark mathematical functions.

Keywords: FPSO+FGA, PSO, GA, SA, PS, Bio-Inspired Optimization Methods.

1 Introduction

We describe in this paper an evolutionary method combining PSO and GA, to give us an improved FPSO+FGA hybrid method. We apply the hybrid method to mathematical function optimization to validate the new approach. In this case, we are using a set of mathematical benchmark functions [4][5][13][17] to compare the optimization results among a GA,PSO,SA, GPS and the proposed method FPSO+FGA.

Several approaches have been proposed for PSO and GA, for example, in [15] can be seen an approach with GA and PSO for control vector for loss minimization of induction motor. In [16] it can be seen an approach with PSO, GA and Simulated Annealing (SA), for scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. Also, we compared the experimental results obtained in this paper with the results obtained in [17]. Also, in [19][22] a similar approach is shown.

The main motivation of this method is to combine the characteristics of a GA and PSO [1][2]. We are using several fuzzy systems to perform dynamical parameter adaptation. For decision making between the methods depending on the results that we are generating we are using another fuzzy system. The paper is organized as follows: in section 2 a description of the optimization methods used in this paper are presented, in section 3 the proposed method FPSO+FGA, mathematical description and the fuzzy systems are described, in section 4 the experimental results are described, and finally in section 5 the conclusions obtained after the study of the proposed evolutionary computing methods are presented.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 131–142, 2011.

2 Optimization Methods

2.1 Genetic Algorithms

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 Systems* [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 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].

2.2 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* [11].

2.3 Simulated Annealing

SA is a generic probabilistic metaheuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global optimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive enumeration provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends both on the difference between the corresponding function values and also on a global parameter *T* (called the *temperature*), that is gradually decreased during the process [18].

2.4 Pattern Search

Pattern search is a family of numerical optimization methods that do not require the gradient of the problem to be optimized and PS can hence be used on functions that are not continuous or differentiable. Such optimization methods are also known as direct-search, derivative-free, or black-box methods.

The name, pattern search, was coined by Hooke and Jeeves [20]. An early and simple PS variant is attributed to Fermi and Metropolis when they worked at the Los Alamos National Laboratory as described by Davidon [21] who summarized the algorithm as follows:

They varied one theoretical parameter at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, they halved the step size and repeated the process until the steps were deemed sufficiently small.

3 FPSO+FGA Method

The general approach of the proposed method FPSO+FGA can be seen in Figure 1. The method can be described as follows:

1. It receives a mathematical function to be optimized
2. It evaluates the role of both GA and PSO.
3. A main fuzzy system is responsible for receiving values resulting from step 2.
4. The main fuzzy system decides which method to use(GA or PSO)
5. Another fuzzy system receives the Error and DError as inputs to evaluates if is necessary change the parameters in GA or PSO.
6. There are 3 fuzzy systems. One is for decision making (is called main fuzzy), the second one is for changing parameters of the GA (is called fuzzyga) in this case change the value of crossover (k_1) and mutation (k_2) and the third fuzzy system is used to change parameters of the PSO (is called fuzzypso) in this case change the value of social acceleration (c_1) and cognitive acceleration (c_2).

7. The main fuzzy system decides in the final step the optimum value for the function introduced in step 1. Repeat the above steps until the termination criterion of the algorithm is met.

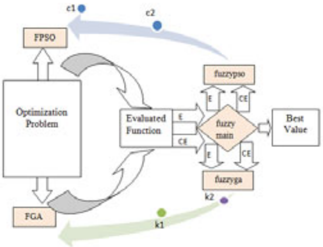
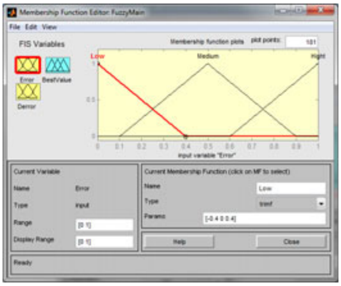


Fig. 1. The FPSO+FGA scheme

The basic idea of the FPSO+FGA scheme is to combine the advantages of the individual methods using a fuzzy system for decision making and the others two fuzzy systems to improve the parameters of the FGA and FPSO when is necessary.

As can be seen in the proposed hybrid FPSO+FGA method, it is the internal fuzzy system structure, which has the primary function of receiving as inputs (Error and DError) the results of the FGA and FPSO outputs. The fuzzy system is responsible for integrating and decides which are the best results being generated at run time of the FPSO+FGA. It is also responsible for selecting and sending the problem to the "fuzzypso" fuzzy system when the FPSO is activated or to the "fuzzyga" fuzzy system when FGA is activated. Also activating or temporarily stopping depending on the results being generated. Figure 2 shows the membership functions of the main fuzzy system that is implemented in this method. The fuzzy system is of Mamdani type because it is more common in this type of fuzzy control and the defuzzification method is the centroid. In this case, we are using this type of defuzzification because in other papers we have achieved good results with it [4]. The membership functions are of triangular form in the inputs and outputs as is shown in Figure 2. 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 Low and DError is Low then best value is Good (view Figure 3). Figure 4 shows the fuzzy system rule viewer. Figure 5 shows the surface corresponding to this fuzzy system. The other two fuzzy systems are similar to the main fuzzy system.



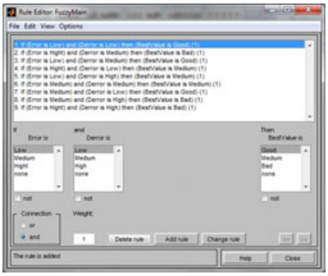
plot points: 101

Current Variable
Name Error
Type input
Range [0 1]
Display Range [0 1]

Current Membership Function (click on MF to select)
Name Low
Type trimf
Params [-0.4 0 0.4]

Ready

Fig. 2. Membership functions of the fuzzy system



1. If (Error is Low) and (Derror is Low) then (BestValue is Good) (1)
2. If (Error is Hight) and (Derror is Medium) then (BestValue is Bad) (1)
3. If (Error is Low) and (Derror is Medium) then (BestValue is Good) (1)
4. If (Error is Hight) and (Derror is Low) then (BestValue is Medium) (1)
5. If (Error is Low) and (Derror is High) then (BestValue is Medium) (1)
6. If (Error is Medium) and (Derror is Medium) then (BestValue is Medium) (1)
7. If (Error is Medium) and (Derror is Low) then (BestValue is Good) (1)
8. If (Error is Medium) and (Derror is High) then (BestValue is Bad) (1)
9. If (Error is Hight) and (Derror is High) then (BestValue is Bad) (1)

Fig. 3. Rules of the fuzzy system

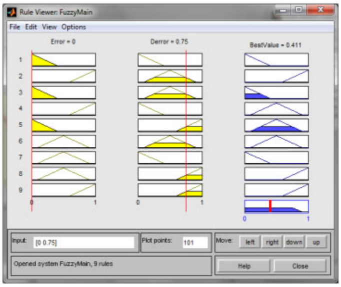


Fig. 4. Rule viewer for the fuzzy system

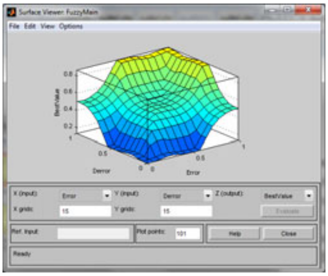


Fig. 5. Surface of the main fuzzy system

4 Experimental Results

To validate the proposed method we used a set of 5 benchmark mathematical functions; all functions were evaluated with different numbers of dimensions, in this case, the experimental results were obtained with 32, 64 and 128 dimensions.

Table 1 shows the definitions of the mathematical functions used in this paper. The global minimum for the test functions is 0.

Table 1. Mathematical functions

Function	Definition
De Jong's	$f_1 = \sum_{n=i}^{N} x_n^2$
Rotated Hyper- Ellipsod	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$
Rosenbrock's Valley	$f(x) = \sum_{i=1}^{n-1} 100 \left(x_{i+1} - x_i^2 \right)^2 + (1 - x_i)^2$
Rastrigin's	$f(x) = 10.n + \sum_{i=1}^{n} \left(x_i^2 - 10.\cos(2\pi x_i) \right)$
Griewank's	$f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$

Tables 2, 3 and 4 show the experimental results for the benchmark mathematical functions used in this research with the proposed method FPSO+FGA. The Tables show the experimental results of the evaluations for each function with 32, 64 and 128 dimensions; where it can be seen the best and worst values obtained, and the average of 50 times after executing the method.

Table 2. Experimental results with 32 dimensions

Function	Average	Best	Worst
De Jong's	7.73E-28	1.08E-29	1.093E-17
Rotated Hyper- Ellipsod	1.07E-18	3.78E-20	6.19E-13
Rosenbrock's Valley	0.000025	0.000006	0.0516
Rastrigin's	9.68E-15	2.54E-15	3.64E-14
Griewank's	2.41E-12	4.25E-13	9.98E-10

Table 3. Experimental results with 64 dimensions

Function	Average	Best	Worst
De Jong's	6.75E-25	2.10E-27	1.093E-15
Rotated Hyper- Ellipsod	3.09E-15	4.99E-17	6.19E-10
Rosenbrock's Valley	0.00325	0.000621	0.0416
Rastrigin's	0.00332	0.000310	8.909
Griewank's	0.001987	0.000475	10.02

Table 4. Simulations results with 128 dimensions

Function	Average	Best	Worst
De Jong's	1.68E-21	1.00E-23	2.089
Rotated Hyper- Ellipsod	3.09E-12	4.99E-15	8.09
Rosenbrock's Valley	0.299	0.00676	9.0456
Rastrigin's	0.256	0.0543	10.098
Griewank's	0.1987	0.0475	12.98

Also, to validate our approach several test were made with the GA, PSO, SA and PS optimization methods. Tables 5, 6 and 7 show the experimental results with the GA methods.

Table 5. Experimental results with 32 dimensions with GA

Function	Average	Best	Worst
De Jong's	0.00094	1.14E-06	0.0056
Rotated Hyper- Ellipsod	0.05371	0.00228	0.53997
Rosenbrock's Valley	3.14677173	3.246497	3.86201
Rastrigin's	82.35724	46.0085042	129.548
Griewank's	0.41019699	0.14192331	0.917367

Table 6. Experimental results with 64 dimensions with GA

Function	Average	Best	Worst
De Jong's	0.00098	1.00E-05	0.0119
Rotated Hyper- Ellipsod	0.053713	0.00055	0.26777
Rosenbrock's Valley	3.86961452	3.51959	4.153828
Rastrigin's	247.0152194	162.434	347.2161
Griewank's	0.98000573	0.78743	1.00242

Table 7. Simulations results with 128 dimensions with GA

Function	Average	Best	Worst
De Jong's	9.42E-04	1.00E-05	0.0071
Rotated Hyper- Ellipsod	0.05105	0.000286	0.26343
Rosenbrock's Valley	4.2099029	3.8601773	4.558390
Rastrigin's	672.6994	524.78094	890.93943
Griewank's	1.0068884	1.0051	1.00810

In Tables 8, 9 and 10 We Can Appreciate the Experimental Results with PSO.

Table 8. Experimental results with 32 dimensions with PSO

Function	Average	Best	Worst
De Jong's	5.42E-11	3.40E-12	9.86E-11
Rotated Hyper- Ellipsod	5.42E-11	1.93E-12	9.83E-11
Rosenbrock's Valley	3.2178138	3.1063	3.39178762
Rastrigin's	34.169712	16.14508	56.714207
Griewank's	0.0114768	9.17E-06	0.09483

Table 9. Experimental results with 64 dimensions with PSO

Function	Average	Best	Worst
De Jong's	4.89E-11	2.01E-12	9.82E-11
Rotated Hyper- Ellipsod	6.12E-11	5.95E-12	9.91E-11
Rosenbrock's Valley	3.3795190	3.227560	3.5531097
Rastrigin's	126.01692	72.364868	198.1616
Griewank's	0.3708721	0.137781	0.667802

Table 10. Experimental results with 128 dimensions with PSO

Function	Average	Best	Worst
De Jong's	5.34E-11	3.323E-12	9.73E-11
Rotated Hyper- Ellipsod	8.60E-11	2.004E-11	9.55E-11
Rosenbrock's Valley	3.6685710	3.5189764	3.8473198
Rastrigin's	467.93181	368.57558	607.87495
Griewank's	0.9709302	0.85604	1.00315

In Tables 11, 12 and 13 we can appreciate the experimental results with the SA Method.

Table 11. Experimental results with 32 dimensions with SA

Function	Average	Best	Worst
De Jong's	0.1210	0.0400	1.8926
Rotated Hyper- Ellipsod	0.9800	0.0990	7.0104
Rosenbrock's Valley	1.2300	0.4402	10.790
Rastrigin's	25.8890	20.101	33.415
Griewank's	0.9801	0.2045	5.5678

Table 12. Experimental results with 64 dimensions with SA

Function	Average	Best	Worst
De Jong's	0. 5029	0.0223	1.8779
Rotated Hyper- Ellipsod	6.0255	3.1667	22.872
Rosenbrock's Valley	5.0568	3.5340	7.7765
Rastrigin's	81.3443	50.9766	83.9866
Griewank's	1.9067	0.9981	6.3561

Table 13. Simulations results with 128 dimensions with SA

Function	Average	Best	Worst
De Jong's	0.3060	0.2681	3.089
Rotated Hyper- Ellipsod	5.0908	3.4599	85.09
Rosenbrock's Valley	8.0676	2.9909	9.0456
Rastrigin's	180.4433	171.0100	198.098
Griewank's	4.3245	1.5567	12.980

In Tables 14, 15 and 16 we can appreciate the experimental results with the PS Method.

Table 14. Experimental results with 32 dimensions with PS

Function	Average	Best	Worst
De Jong's	0. 3528	0.2232	2.0779
Rotated Hyper- Ellipsod	16.2505	3.1667	25.782
Rosenbrock's Valley	4.0568	3.0342	5.7765
Rastrigin's	31.4203	25.7660	33.9866
Griewank's	0.6897	0.0981	3.5061

Table 15. Simulations results with 64 dimensions with PS

Function	Average	Best	Worst
De Jong's	1.0034	0.9681	1.890
Rotated Hyper- Ellipsod	20.0908	4.5099	35.090
Rosenbrock's Valley	9.6006	5.9909	11.562
Rastrigin's	53.3543	50.0100	55.098
Griewank's	3.2454	0.5647	6.9080

Table 16. Simulations results with 128 dimensions with PS

Function	Average	Best	Worst
De Jong's	4.0034	1.9681	9.9320
Rotated Hyper- Ellipsod	32.0908	9.5099	37.090
Rosenbrock's Valley	12.6980	8.0887	17.234
Rastrigin's	74.5043	60.1100	80.098
Griewank's	9.0771	5.6947	20.0380

4.1 Statistical Test

To validate this approach we performed a statistical test with the analyzed methods. The test used for these experiments was the T-Student test.

In table 17, we can see the test for FPSO+FGA vs GA.

Where: T Value = -1.01, P Value = 0.815.

Table 17. Two-sample T-Test for FPSO+FGA vs GA

Method	Mean	StDev	SE Mean
FPSO+FGA	0.0217	0.0269	0.012
GA	106	234	105

In table 18, a T- test between the proposed method vs SA is shown. Where: T Value = -1.06, P Value = 0.826.

Table 18. Two-sample T-Test for FPSO+FGA vs SA

Method	Mean	StDev	SE Mean
FPSO+FGA	0.0217	0.0269	0.012
SA	35.9	75.6	34

In table 19, a T- test between the GA vs PSO is shown. Where: T Value = 0.37, P Value = 0.64.

Table 19. Two-sample T-Test for GA vs PSO

Method	Mean	StDev	SE Mean
GA	50	138	36
PSO	32	121	31

We can see after applying the T-Student test with the analyzed methods, how the proposed method is better than other methods used in this research, because, for example, the T- test shown in Table 19, between GA and PSO the difference is very small. However, with the proposed method compared with other approaches the difference is good statistically speaking.

In table 20 we can see a comparison of results among the used methods in this paper with the five mathematical functions evaluated for 128 variables.

Table 20. Comparison results among the used methods with 128 variables

Function	FPSO+FGA	GA	PSO	SA	PS
De Jong's	1.00E-23	1.00E-05	3.32E-12	0.2681	1.9681
Rotated Hyper-Ellipsod	4.99E-15	0.000286	2.00E-11	3.4599	9.5099
Rosenbrock's Valley	0.00676	3.8601773	3.5189764	2.9909	8.0887
Rastrigin's	0.0543	524.78094	368.57558	171.01	60.11
Griewank's	0.0475	1.0051	0.85604	1.5567	5.6947

Figure 6 shows graphically the comparison seen in table 20. In this figure we note that the difference among the best objective values obtained, for example, the proposed method (FPSO+FGA) with 128 variables was able to optimize the five functions, and the other analyzed methods only with some functions were to able to obtain good results.

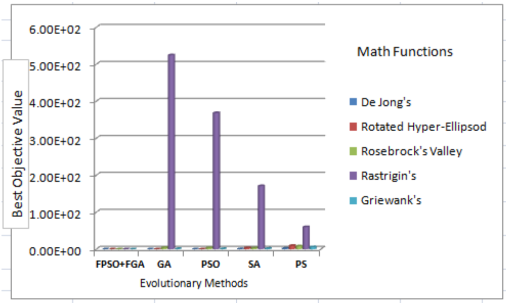


Fig. 6. Comparison results among the used methods

5 Conclusions

The analysis of the experimental results of the bio inspired method considered in this paper, the FPSO+FGA, lead us to the conclusion that for the optimization of these benchmark mathematical functions this method is a good alternative, because it is easier and very fast to optimize and achieve good results than to try it with PSO, GA and SA separately [5], especially when the number of dimensions is increased. This is, because the combination of PSO and GA with fuzzy rules allows adjusting the parameters in the PSO and GA. Also, the experimental results obtained with the proposed method in this research were compared with other similar approaches [17], achieving good results.

References

1. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Designs. Springer, Heidelberg (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, pp. 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. In: The Emerging Science of Artificial Life, p. 114. Princeton University Press (1994)
10. Valdez, F., Melin, P.: Parallel Evolutionary Computing using a cluster for Mathematical Function Optimization, Nafips, San Diego CA, USA, pp. 598–602 (June 2007)
11. Angeline, P.J.: Using Selection to Improve Particle Swarm Optimization. In: Proceedings 1998 IEEE World Congress on Computational Intelligence, Anchorage, Alaska, pp. 84–89. IEEE (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. Kim, D., Hirota, K.: Vector control for loss minimization of induction motor using GA–PSO. Applied Soft Computing 8, 1692–1702 (2008)
16. Liu, H., Abraham, A.: Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm.Article in press, Future Generation Computer Systems
17. Mohammed, O., Ali, S., Koh, P., Chong, K.: Design a PID Controller of BLDC Motor by Using Hybrid Genetic-Immune. Modern Applied Science 5(1) (February 2011)
18. Kirkpatrick, S., Gelatt, C.J., Vecchi, M.: Optimization by Simulated Annealing. Science 220(4598), 671–680 (1983)

19. 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)
20. Hooke, R., Jeeves, T.A.: 'Direct search' solution of numerical and statistical problems. Journal of the Association for Computing Machinery 8(2), 212–229 (1961)
21. Davidon, W.C.: Variable metric method for minimization. SIAM Journal on Optimization 1(1), 1–17 (1991)
22. Ochoa, A., Ponce, J., Hernández, A., Li, L.: Resolution of a Combinatorial Problem using Cultural Algorithms. JCP 4(8), 738–741 (2009)

Fundamental Features of Metabolic Computing

Ralf Hofestädt

Bielefeld University, AG Bioinformatics and Medical Informatics, Bielefeld
hofestae@techfak.uni-bielefeld.de

Abstract. The cell is the basic unit of life and can be interpreted as a chemical machine. The present knowledge of molecular biology allows the characterization of the metabolism as a processing unit/concept. This concept is an evolutionary biochemical product, which has been developed over millions of years. In this paper we will present and discuss the analyzed features of metabolism, which represent the fundamental features of the metabolic computing process. Furthermore, we will compare this molecular computing method with methods which are defined and discussed in computer science. Finally, we will formalize the metabolic processing method.

Keywords: Metabolic Computing, Metabolic Features, Genetic Grammar, Language of Life.

1 Introduction

The global goal of computer science is to develop efficient hard- and software. The computer scientist tries to do this exercise on different levels: technology (ULSI, biochips …), computer architectures (data flow computer, vector machine, …), supercompilers, operating systems (distributed) and programming languages (Occam, Par-C, …). Different processing methods are already discussed in the field of theoretical computer science: probabilistic algorithms, stochastic automaton, parallel algorithms (parallel random access machine, uniform circuits) and dynamic automata (hardware modification machine). Furthermore, the discussion of adaptive algorithms is of great interest. However, the speed-up value of parallel architectures including new software and new technologies cannot be higher than linear. Overall, computer scientists have to develop new and powerful processing and computational methods. Therefore, the study of natural adaptive algorithms is one fundamental innovation process over the last years. Regarding the literature, we can see that the metabolic computational method has not been discussed until now. This is the topic of this paper. Therefore, we will present the analyzed features of metabolism, which are responsible for the biochemical processes inside the living cell. Furthermore, we will interpret the cell as a chemical machine [1] and develop a grammatical formalism of metabolic computing.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 143–152, 2011.

2 Features

Since 1944 it is known that the deoxyribonucleic acid (DNA) controls metabolism. Watson and Crick introduced their model of DNA in the year 1953. Since then complex metabolic processes have been analyzed. The model of gene regulation from Jacob and Monod is still the fundamental contribution [2]. Today, the methods of molecular biology allow isolating, sequencing, synthesizing and transforming DNA-structures. Based on these technologies and the internet, more than 1000 molecular databases are available worldwide.

Nowadays it is well-known that the analyzed DNA-structures control metabolism indirectly. DNA controls metabolism using special proteins (enzymes) which catalyse biochemical processes. Therefore, enzymes represent the biosynthetic products of structure genes, which can be interpreted as genetic instructions. The DNA-structures represent the minimal structure elements of a programming language (datatype, operation, control structure and punctuation). Furthermore, molecular experiments have reinforced the view that DNA-structures can be interpreted as a programming language [3, 1]. Moreover, the analysis of the DNA-structures pointed out complex language constructs [4]:

1. Parallel Computation (**PC**)
   Genetic instructions (structure genes) can be activated simultaneously. Therefore, based on the concentration of the enzyme RNA Polymerase and other molecular components, different structure genes can start the transcription and translation process.
2. Probabilistic Computation (**PrC**)
   The transcription process of a structure gene depends on the so called Pribnow box, which specifies the probability of the transcription process. Therefore, the probabilistic activation of genetic instructions is characteristic.
3. Variable Granulation (**VG**)
   The number of simultaneously activated genetic instructions depends on several factors (fundamental is the concentration of RNA-polymerase, t-RNA-structures and ribosomes etc.).
4. Dynamic Genome (**DG**)
   The genome is a dynamic structure, because mutation, virus-DNA (-RNA), and transposons are able to modify the genome.
5. Modular Organization (**MO**)
   The genome is organized by modules, because homeotic genes are able to control gene batteries.
6. Overlapping Genes (**OG**)
   Virus RNA shows that the DNA can be read from both sides and that genes can overlap.
7. Data Flow and Control Flow (**DF, CF**)
   The v. Neumann computer defines the control flow architecture, which means that the address of the next instruction is represented by the program counter. The dataflow concept says that any instruction will be executed in the case that all operands are available.

Furthermore, the genetic memory is not a random access memory. More or less every cell of an organism represents the whole genome and most of the genome structures are evolutionarily redundant. To clarify how far machine and language models represent the analyzed metabolic characteristics it is necessary to discuss well-known machine and language models. It is not possible to put through a complete discussion, so the discussion will be restricted to well-known models.

Table 1. Machine and language models and their characteristics

Model/Characteristics	PC	DG	PrC	VG	MO	OG	DF	CF
one-tape Turing-Machine (TM) [5]	no	no	no	no	yes	no	yes	no
Probabilistic Turing-Machine [6]	no	no	yes	no	yes	no	yes	no
Random Access Machine [7]	no	no	no	no	yes	no	no	yes
Parallel RAM (PRAM) [8]	yes	no	no	yes	yes	no	no	yes
Cellular Automata [9]	yes	no	no	no	no	no	yes	no
Uniform Circuits [10]	yes	no	no	no	no	no	yes	no
Vector Machine [11]	yes	no	no	no	yes	no	no	yes
Hadware Modification Machine [12]	yes	yes	no	yes	no	no	yes	no
Classifier Machine [13]	yes	no	no	no	no	no	yes	no
While Program [14]	no	no	no	no	yes	no	no	yes
Chomsky Grammars [5]	no	no	no	no	no	no	yes	no
Lindenmayer-System [15]	yes	yes	no	no	no	no	yes	no

The characteristics of metabolic processing, which are the basic elements of biological complexity, do not disrupt the well-known methods in computer science. However, we have to consider that our knowledge of gene regulation and the semantics of some analyzed DNA-structures is still rudimentary.

Furthermore, table 1 shows that no theoretical model exists in computer science which represents and includes all metabolic features. The integration of these elements into a model will represent the biological computation model.

3 Genetic Grammar

Table 1 shows the characteristics of metabolic processing. A method which embraces the metabolic features will expand the frame of methods which are discussed in computer science. In this paper we choose a grammatical formalism to define the genetic language. The basis of this formalism is the semi-Thue system which will be extended including the presented metabolic features.

Definition 1
Let Σ be a finite alphabet and $n \in IN^+$. $m \in \Sigma^n$ is called a ***message***.

(1)

Definition 2
Let Σ be a finite alphabet, $n \in IN^+$ and $\Gamma = \Sigma \cup \{\#\}$. A tuple $c = (\alpha, ß)$ with $\alpha \in \Gamma^n$ (precondition) and $ß \in \Sigma^n$ (postcondition) is called an ***n-rule***. The set $C_n = \{ c : c = (\alpha,ß)$ is an n-rule $\}$ denotes the set of all n-rules.

(2)

Definition 3
Let $\alpha \in \Sigma^n$ and $ß \in \Gamma^n$ with $n \in IN^+$. α is ***similar*** to ß, in symbols $\alpha \approx ß$, iff

$$\forall\ i \in \{1,...,n\}\ \ \alpha_i = ß_i \vee ß_i = \# \vee \alpha_i = \#.$$

(3)

Definition 4
The 4-tuple $(n, \Sigma, \Phi, \Theta)$ with $\in IN^+$, Σ a finite alphabet, $\Phi \subseteq C_n$ a set of n-rules and $\Theta \subseteq \Sigma^n$ the start message set is called ***basic system***.

(4)

The working method of this system will be defined.

Definition 5
Let $G = (n, \Sigma, \Phi, \Theta)$ be a basic system and $D \subseteq \Sigma^n$. Any rule $c = (\alpha, ß) \in \Phi$ is ***activated*** by the message set D, in symbols c(D), iff $\exists\ m \in D\ \ m \approx \alpha$.

$\Phi(D) = \{ c \in \Phi : c$ is activated $\}$ denotes the set of all activated n-rules.

(5)

Any activated n-rule can go into action.

Definition 6
Let $G = (n, \Sigma, \Phi, \Theta)$ be any basic system and $D \subseteq \Sigma^n$, $c \in \Phi$, $m \in D$ and $ß \in \Sigma^n$. (m, ß) is called ***action*** of n-rule c, in symbols $m\ {}_c\!\!-\!\!> ß$, iff $c = (\alpha, ß)$ and $m \approx \alpha$.

(6)

The simultaneous action of all activated n-rules will be called one-step derivation.

Definition 7
Let $G = (n, \Sigma, \Phi, \Theta)$ be any basic system and $D \subseteq \Sigma^n$. D is called ***one-step derivation*** into D', in symbols $D => D'$, iff $D' \subseteq \{ ß \in \Sigma^n : \exists\ m \in D\ \ \exists\ c = (\alpha,ß) \in \Phi\ \ m\ {}_c\!\!-\!\!> ß \}$.

(7)

Definition 8

Let $G = (n, \Sigma, \Phi, \Theta)$ be any basic system and $D_i \in \Sigma^n$ for $i = 0,...,k$ with $k \in IN^+$. $(D_0,..,D_k)$ is called ***derivation***, iff $\forall\, i \in \{1,...,k\text{-}1\}\ D_i => D_{i+1}$. For a derivation D into D' we write in symbols $D\ {}_k\!\!=> D'$.

(8)

Based on this formal description we can define the language.

Definition 9

Let $G = (n, \Sigma, \Phi, \Theta)$ be any basic system. $L(G) = \{\ \varsigma \in M : \Theta\ {}_*\!\!=> \varsigma\ \}$ is called language of G.

(9)

The probability feature is the first extension of the basic system.

Definition 10

Any 5-tuple $(n, \Sigma, \Phi, \Theta, \delta)$ with $G = (n, \Sigma, \Phi, \Theta)$ is basic system and $\delta: \Phi \text{ -> } [0,1]_Q$ a total function is called a ***probability basic system*** and $\delta(c)$ is called ***action probability*** of $c \in \Phi$.

(10)

The action probability can be interpreted as follows:

if message m activates n-rule c, then the probability of the event "c will occur in action by m" is $\delta(c)$. If there are various messages $m_1,...,m_k$ which can activate the same n-rule

$$c = (\alpha,ß)\ (m_1 \approx \alpha,\ m_2 \approx \alpha,...,m_k \approx \alpha), \tag{11}$$

then all events "c will occur in action by m_i" will be independent.

For any probability basic system

$$G = (n, \Sigma, \Phi, \Theta, \delta) \tag{12}$$

A is called derivation, iff A is a derivation in the basic system. For each derivation the probability can be evaluated. Firstly, we can evaluate the probability $P(N' \mid N)$ to transform the message N into the message N' in the next generation. Therefore, we consider any message set $N \subseteq \Sigma^n$ and pairs (m,c) with $m \in N$, $c \in \Phi$ and c is activated by m. Let $(m_1,c_1),...,(m_k,c_k)$ be such pairs in any order (lexicographical order) and k its quantity. Every word $w \in \{\ L,R\ \}^k$ denotes a set of events which describes a transformation into a new message set (one-step derivation).

Let be

$$w = a_1a_2..a_k. \tag{13}$$

w corresponds to the event:

for i = 1..k,
c_i will ***occur*** in action by message m_i, if $a_i = L$
c_i will ***not occur*** in action by message m_i, if $a_i = R$

these are independent events and the probability of the one-step derivation is:

$$P(W) ::= \prod_{i=1..k} q_i \qquad \text{with } q_i = \delta(c_i)\ (1-\delta(c_i)) \text{ if } a_i = L\ (a_i = R). \tag{14}$$

Each event w will produce an output message h(w). This is the set of post-conditions of the n-rules which will be in action:

$$h(a_1..a_k) = \{\ ß : \exists\ i \in \{\ 1,...,k\ \}\ \ a_i = L \text{ and } c_i = (\alpha,ß)\ \}. \tag{15}$$

The sum of all probabilities of events w which produces output message h(w) is equal to N' and denotes the probability of the message transformation N into N'.

$$P(N'|N) = \ \Sigma_{h(w)=N'}\ P(w) \tag{16}$$

In the next step we define a new class of rules which will allow control of probability values. Moreover, all rules will be extended by visibility flags, so that every rule is visible or invisible. To control these flags it is necessary to define one more class of rules.

Definition 11

Let $n \in N$, Σ be a finite alphabet with $\# \notin \Sigma$ and $\Gamma \in \Sigma \cup \{\#\}$. A 2-tuple $(\alpha,ß)$ is called ***n-message rule*** with precondition $\alpha \in \Gamma^n$ and post-condition $ß \in \Sigma^n$. A 3-tuple $(\alpha,ß,a)$ is called n-regulation rule with pre-condition $\alpha \in \Gamma^n$, target domain $ß \in \Sigma^n$ and regulator $a \in \{+,-\ \}$. A 3-tuple $(\alpha,ß,p)$ is called n-probability rule with pre-condition $\alpha \in \Gamma^n$, target domain $ß \in \Sigma^n$ and the change $a \in [0,1]_Q$.

c is called n-rule, iff c is n-message rule or n-regulation rule or n-probability rule. (17)

Now we are able to define the genetic grammar.

Definition 12

Let $n \in IN$, Σ a finite alphabet with $\# \notin \Sigma$, Φ a set of n-rules, Θ_0 a start message set,

B_0: Φ -> { +,- } a total function and δ_0: Φ -> $[0,1]_Q$ a total function. A 6-tuple G = $(n, \Sigma, \Phi, \Theta_0, B_0, \delta_0)$ is called genetic grammar with message length n, message alphabet Σ and rule set Φ. Φ_N, Φ_R and Φ_P denotes the set of message rules, regulation rules and probability rules of Φ.

(18)

Furthermore the configuration of a genetic grammar is important.

Definition 13

Let G = $(n, \Sigma, \Phi, \Theta_0, B_0, \delta_0)$ be any genetic grammar. A triple (N, B, δ) with $N \in \Sigma^n$, B: Φ -> { +,- } a total function and δ: Φ -> $[0,1]_Q$ a total function is called ***configuration*** of the genetic grammar G with message set N, visibility B and rule probability δ. $(\Theta_0, B_0, \delta_0)$ is called start configuration. Notation:

$$S = \{\ B : \Phi \text{ -> } \{\ +,-\ \} \text{ a total function } \} \text{ and } R = \{\ \delta : \Phi \text{ -> } [0,1]_Q \text{ a total function } \} \tag{19}$$

Any n-rule $c \in \Phi$ is visible (invisible), iff B(c) = '+' (B(c) = '-'). For any n-rule c B(c) is called visibility and δ(c) the action probability of c. An n-rule is activated in any configuration (N, B, δ), iff it is visible and there is a message in the set N which is similar to the precondition of this rule. Any activated rule will occur in action by its rule probability (corresponding to the rule probabilities). The origin of a message is the effect of an action of a special message rule (the same effect as in the probability basic system).

The action of a regulation rule can change the visibility of other rules: if the message is in the target domain of a regulation rule r similar to a precondition of a rule c' $\in \Phi$ and the visibility of rule c' is not equal to the regulator of rule r, then the regulator will be the new visibility of c'. This means, regulation '+' will change from visible to invisible and regulation '-' will change from invisible to visible. It is possible that various regulation rules will influence the visibility of a rule. In this case, the visibility will change as described above.

The action of a probability rule can change the probability of other rules: if the target domain of a probability rule is similar to the pre-condition of a rule c' $\in \Phi$, then the change of rule r will be the new probability of c'. It is possible that various probability rules will influence the probability of one rule. In this case, the change will be the maximum of all changes which are possible in this state.

The configuration (N, B, δ) will be transformed into configuration (N', B', δ'), iff the action of a subset of the activated rules will be produce N', B' und δ' (visibilities and probabilities which would not be modified will be unchanged).

It is possible to define various languages which represent different points of view.

$$L(G,i) = \{ N \subseteq \Sigma^n : \exists B \in S, \delta \in R \text{ with } (\Theta_0, B_0, \delta_0) {}_i\!\!=> (N, B, \delta) \} \tag{20}$$

$$L(G) = \{ N \subseteq \Sigma^n : \exists B \in S, \delta \in R \text{ with } (\Theta_0, B_0, \delta_0) {}_*\!\!=> (N, B, \delta) \} \tag{21}$$

$$L_s(G,i) = \{ M : P_K(M,i) = s \} \tag{22}$$

$$L_s(G) = \{ M : \exists i \in IN \ \ P_K (M,i) = s \} \tag{23}$$

Moreover, there are well-known metabolic processes (mutation and genetic operators) which cannot be described by any rules. These metabolic phenomena only occur rarely so it isn't possible to take these phenomena into the grammatical formalism.

4 Metabolic System

A cell is a chemical machine based on biochemical reactions. Metabolism is based on a non-deterministic method which leads to a wide spectrum of possible metabolic reactions. However, a genetic grammar can be interpreted as a procedure which solves a special problem. The evolution of a cell is based on strategies as mutation, selection and genetic operations which are called modification processes. A genetic grammar is called a metabolic system, iff the one-step derivation is extended by the modification process. The derivation of a metabolic system is called metabolic computation. A metabolic system which has a start configuration

$$K_0 = (\Theta_0, B_0, \delta_0) \tag{24}$$

will terminate, iff there exist a metabolic computation which will lead to a configuration

$$K_n = (\Theta_n, B_n, \delta_n) \tag{25}$$

and there is no activated rule in Θ_n. In this case the message set Θ_n is called the solution of the metabolic computation by input Θ_0. Metabolic systems differ in comparison with genetic algorithms because the metabolic system is a procedure which solves a special exercise and not a problem class. Moreover, metabolic systems expand the classical algorithm method: data flow control, modification of data and rules, the metabolic computation is not definite and parallel computation and termination is not uncertain.

5 Hardware Concept - Complexity

In the following, the discussion is restricted to the activation of the genetic grammar because this is the kernel unit. Moreover, we begin with a few naive assumptions: there are no problems in timing, there are ideal circuits (AND-gates, OR-gates with unlimited fan-in and fan-out) and the consumption of energy will not be considered. The message store holds the actual messages. This will be a special memory unit which is able to read and write all words simultaneously. A 'quasi' associative memory represents n-rules. Here, any word represents the pre-condition (the first n bits) and the post-condition (the last n bits) of an n-rule. Every pre-condition of the associative memory is attached to a mask register. In this way it is possible to mask every pre-condition. This represents an extension of the alphabet. Furthermore, every word of the associative memory is coupled with a visibility flag (flip-flop) and a probability value (register of the length k - N). All probability values are stored in a separate probability memory. This naive realization is based on a random generator which produces bit strings of length

$$k*o\ (\{\ 0,1\ \}^h \text{ with } h = k*o). \tag{26}$$

A bit string will divide into o substrings of length k. Consequently every probability register will couple with a substring. The comparison between the substring and the contents of the probability register is the basis for the evaluation of the specific probability flag: example:

```
i = 1..o
      IF prob.-value(i)= value of the substring
        THEN
            Probability flag(i) = 1
       ELSE  probability flag = 0
```

The logic unit which realizes the activation of the genetic grammar consists of m * o logic units ($m,o \in IN^+$).

With the assumption that the random generator will produce random strings after the run time of two gates the realization of activity will use a run time of four gates. The resources for the logic unit are assuming that the fan-in and fan-out of each gate is unlimited the logic unit requires the following:

$$(m * o) * (3n + 1) + o \tag{27}$$

gates and

$$(8n + 3) * (m * o) + o \tag{28}$$

wires.

The integration of the modification process will require more hardware which will extend the complexity of the metabolic system.

6 Discussion

Computer scientists have to join new processing methods and new architectures, which will expand the linear speed-up. Attention has to be given to processing methods of biological systems because such systems are able to solve hard problems. The well-discussed methods of neural networks and genetic algorithms are based on these ideas, because macroscopic characteristics of the natural processing have been transformed into the theory of algorithms. Generally, in this paper we discuss the microscopic dimension of natural processing for the first time. The semi-Thue system was extended step-by-step by the analyzed features of metabolic processing. This formalism is called genetic grammar and allows the definition of metabolic systems [1]. These systems represent metabolic processing methods which have been developed over millions of years by evolutionary processes. This system allows the discussion of the gene regulation phenomena. Chapter 5 shows that large metabolic systems are currently only realizable as software simulations. Our simulation system, which needs to be implemented, will allow the simulation of metabolic processes and the first discussion of metabolic processing.

The developed metabolic system shows that the power of biological systems is based on controlled correlation of: data flow, associative, probabilistic and dynamic data processing.

References

1. Hofestädt, R.: DNA-Programming Language of Life. HBSO 13, 68–72 (2009)
2. Jacob, F., Monod, J.: Genetic regulatory mechanisms in the synthesis of proteins. J. Mol. Biology 3, 318–356 (1961)
3. Vaeck, M., et al.: Transgenic plants protected from insect attack. Nature 328, 33 (1967)
4. Hofestädt, R.: Extended Backus-System for the representation and specification of the genome. Journal of Bioinformatics and Computational Biology 5-2(b), 457–466 (2007)
5. Hopcroft, J.E., et al.: Automate Theory, Languages, And Computation. Addison-Wesley Publishing, Sydny (2009)

6. Gill, J.: Computational Complexity of Probabilistic Turing Machines. SIAM Journal of Computing 6, 675–695 (1977)
7. Aho, A., et al.: The design and analysis of Computer Algorithms. Addison-Wesley Publishing Company, Ontario (2008)
8. Fortune, S., et al.: Parallelism in Random Access Machines. In: Proc. 10th ACM Symposium on Theory of Computing, pp. 114–118 (1978)
9. Vollmer, R.: Algorithmen in Zellularautomaten. Teubner Publisher, Stuttgar (1979)
10. Borodin, A.: On relating time and space to size and depth. SIAM Journal of Computing 6, 733–744 (1977)
11. Pratt, S., et al.: A Characterization of the power of vector machines. Journal of Computer and System Sciences 12, 198–221 (1978)
12. Cook, S.: Towards A Complexity Theory of synchronous Parallel Computation. L'Enseignement Mathematique 27, 99–124 (1981)
13. Burks, A.: The Logic of Evolution. In: Jelitsch, R., Lange, O., Haupt, D., Juling, W., Händler, W. (eds.) CONPAR 1986. LNCS, vol. 237, pp. 237–256. Springer, Heidelberg (1986)
14. Manna, Z.: Mathematical theory of computation. McGraw Hill Publisher, New York (1974)
15. Prusinkiewicz, P., Lindenmayer, A.: The Algorithmic Beauty of Plants. Springer, New York (1990)

Clustering Ensemble Framework via Ant Colony

Hamid Parvin and Akram Beigi

Islamic Azad University, Nourabad Mamasani Branch, Nourabad Mamasani, Iran
hamidparvin@mamasaniiau.ac.ir, beigi@iust.ac.ir

Abstract. Ensemble-based learning is a very promising option to reach a robust partition. Due to covering the faults of each other, the classifiers existing in the ensemble can do the classification task jointly more reliable than each of them. Generating a set of primary partitions that are different from each other, and then aggregation the partitions via a consensus function to generate the final partition, is the common policy of ensembles. Another alternative in the ensemble learning is to turn to fusion of different data from originally different sources. Swarm intelligence is also a new topic where the simple agents work in such a way that a complex behavior can be emerged. Ant colony algorithm is a powerful example of swarm intelligence. In this paper we introduce a new ensemble learning based on the ant colony clustering algorithm. Experimental results on some real-world datasets are presented to demonstrate the effectiveness of the proposed method in generating the final partition.

Keywords: Ant Colony, Data Fusion, Clustering.

1 Introduction

Data clustering is an important technique for statistical data analysis. Machine learning typically regards data clustering as a form of unsupervised learning. The aim of clustering is the classification of similar objects into different cluster, or partitioning of a set of unlabeled objects into homogeneous groups or clusters (Faceli et al., 2006). There are many applications which use clustering techniques to discover structures in data, such as Data Mining (Faceli et al., 2006), pattern recognition, image analysis, and machine learning (Deneubourg et al., 1991).

Ant clustering is introduced by Deneubourg et al. (1991). In that model, the swarm intelligence of real ants is inserted into a robot for the object collecting task. Lumer and Faieta (1994) based on how ants organize their food in their nest, added the Euclidean distance formula as similarity density function to Deneubourg's model. Ants in their model had three kinds of abilities: speed, short-term memory, and behavior exchange.

There are two major operations in ant clustering: picking up an object from a cluster and dropping it off into another cluster (Tsang and Kwong, 2006). At each step, some ants perform pick-up and drop-off based on some notions of similarity between an object and the clusters. Azimi et al. (2009) define a similarity measure based on the co-association matrix. Their approach is fully decentralized and self-organized and allows clustering structure to emerge automatically from the data.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 153–164, 2011.

Liu et al. propose a method for incrementally constructing a knowledge model for a dynamically changing database, using an ant colony clustering. They use information-theoretic metrics to overcome some inherent problems of ant-based clustering. Entropy governs the pick-up and drop behaviors, while movement is guided by pheromones. They show that dynamic clustering can provide significant benefits over static clustering for a realistic problem scenario (Liu et al., 2006).

The rest of the paper is organized as follows: Section 2 considers ant colony clustering. The proposed new space and modified ant clustering algorithm are presented in Section 3 and In Section 4, simulation and results of the clustering algorithm over original feature space versus mapped feature space are discussed. The paper is concluded in Section 5.

2 Ant Colony Clustering

In this section the main aspects of ant colony clustering and its original algorithm is considered. Also some weaknesses of original algorithm are mentioned succinctly and then modeling of this issue is expressed.

2.1 Original Algorithm of Ant Clustering

An original form of ant colony clustering's algorithm includes a population of ants. Each ant operates as an autonomous agent that reorganizes data patterns during exploration to achieve an optimal clustering. Pseudo code of ant colony clustering algorithm is depicted in Algorithm 1.

Objects are represented by the multi-dimensional vector of feature space which is randomly scattered in a 2D space. Ants search the space randomly and they use its short-term memory to jump into a location that is potentially near to an object. They can pick up or drop an object using a probability density obtained by equation 1.

$$f(o_i) = \max\left\{0, \frac{1}{s^2} \sum_{o_j \in Neigh_{s\times s}(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha(1 + \frac{v-1}{v_{\max}})}\right]\right\} \tag{1}$$

Observable local area of an ant that located in room r, is presented by $Neigh_{s\times s}(r)$. Each room including $Neigh_{s\times s}(r)$ and r is a 2D vector. The function d (o_i, o_j) is the distance between two objects o_i and o_j in the original feature space and it is calculated by equation 2. Threshold α is scales the distance between each pair of objects and speed parameter v that control the volume of feature space that an ant explores in each epoch.

$$d(o_i, o_j) = \sqrt{\sum_{k=1}^{m} (o_{ik} - o_{jk})^2} \tag{2}$$

Algorithm 1. Original ant colony clustering

```
Initialize parameter;
For each ant a
    Place random a in one position not occupied by other ants;
For each object o
    Place random o in one position not occupied by other objects;
For t=1 to $t_{max}$
    For each ant a
            g = select a random number uniformly from range [0, 1];
            r = position (a);
            If (loaded (a) and (is-empty(r)))
                    If ($g < p_{drop}$)
                            o = drop (a);
                            Put (r, o);
                            Save (o, r, q);
                    end;
            end;
            Else if (not (loaded (a) or (is-empty(r))))
                    If ($g < p_{pic}$)
                            o = remove (r);
                            pick-up (a, o);
                            search&jump (a, o);
                    end;
            end;
            else
                    Wander (a, v, $N_{dir}$);
            end;
    end;
end;
```

m is the number of original features and o_{ik} is k-th feature of object o_i. Probability that an unloaded ant takes an object that is in the room occupied by the ant, obtained from the equation 3.

$$P_{pick}(o_i) = \left(\frac{k_1}{k_1 + f(o_i)}\right)^2 \tag{3}$$

k_1 is a fixed threshold to control the probability of picking an object. The probability that a loaded ant lays down its object is obtained by equation 4.

$$P_{drop}(o_i) = \begin{cases} 2f(o_i) & if\ \ f(o_i) < k_2 \\ 1 & if\ \ f(o_i) \geq k_2 \end{cases} \tag{4}$$

k_2 is a fixed threshold to control the probability of dropping an object. Similarity measure, speed parameter, local density and short-term memory are described in following.

2.2 Weaknesses of Original Algorithm

The original ant colony clustering algorithm presented above suffers two major weaknesses. First many clusters are produced in the virtual two-dimensional space and it is hard and very time-consuming to merge them and this work is inappropriate.

The second weakness arises where the density detector is the sole measure based on that the clusters are formed in the local similar objects. But it fails to detect their dissimilarity properly. So a cluster without a significant between-object variance may not break into some smaller clusters. It may result in forming the wrong big clusters including some real smaller clusters provided the boundary objects of the smaller clusters are similar. It is because the probability of dropping or picking up an object is dependent only to density. So provided that the boundary objects of the smaller clusters are similar, they placed near to each other and the other objects also place near to them gradually. Finally those small clusters form a big cluster, and there is no mechanism to break it into smaller clusters. So there are some changes on the original algorithm to handle the mention weaknesses.

2.3 Modeling of Ant Colony

In this section some parameters of ant modeling are presented. This parameters are inspired of real-world swarm intelligence.

Perception Area is number of objects that an ant can observe in 2D area s. It is one effective factor to control the overall similarity measure and consequently the accuracy and the computational time of the algorithm. If s is large, it will cause the rapid formation of clusters and therefore generally fewer developed clusters. If s is small, it will cause the slower formation of clusters and therefore the number of clusters will be larger. Therefore, selecting a large value can cause premature convergence of the algorithm, and a small value causes late convergence of the algorithm.

Similarity Scaling Factor (α) is defined in the interval (0, 1]. If α is large, then the similarities between objects will increase, so it is easier for the ants to lay down their objects and more difficult for them to lift the objects. Thus fewer clusters are formed and it will be highly likely that well-ordered clusters will not form. If α is small, the similarities between objects will reduce, so it is easier for the ants to pick up objects and more difficult for them to remove their objects. So many clusters are created that can be well-shaped. On this basis, the appropriate setting of parameter α is very important and should not be data independent.

Speed Parameter (v) can uniformly be selected form range [1, v_{max}]. Rate of removing an object or picking an object up can be affected by the speed parameter. If v is large, few rough clusters can irregularly be formed on a large scale view. If v is small, then many dense clusters can precisely be formed on a small scale view. The speed parameter is a critical factor for the speed of convergence. An appropriate setting of speed parameter v may cause faster convergence.

Short Term Memory mentioned that each ant can remember the original real features and the virtual defined two-dimensional features of the last q objects it drops. Whenever ant takes an object it will search its short term memory to find out which object in the short term memory is similar to the current object. If an object in memory is similar enough to satisfy a threshold, it will jump to the position of the object, hoping the current object will be dropped near the location of the similar object, else if there is no object in memory similar, it will not jump and will hold the object and will wander. This prevents the objects originally belonging to a same cluster to be spitted in different clusters.

Entropy measure is a proper metric in many areas. Combining the information entropy and the mean similarity as a new metric to existing models in order to detect rough areas of spatial clusters, dense clusters and troubled borders of the clusters that are wrongly merged is employed.

Shannon entropy information has been widely used in many areas to measure the uncertainty of a specified event or the impurity of an arbitrary collection of samples. Consider a discrete random variable X, with N possible values $\{x_1, x_2, ..., x_N\}$ with probabilities $\{p(x_1), p(x_2), ..., p(x_N)\}$. Entropy of discrete random variable X is obtained using equation 5.

$$H(X) = -\sum_{i=1}^{N} p(x_i)\log p(x_i) \tag{5}$$

Similarity degree between each pair of objects can be expressed as a probability that the two belong to the same cluster. Based on Shannon information entropy, each ant can compute the impurity of the objects observed in a local area L to determine if the object o_i in the center of the local area L has a high entropy value with group of object o_j in the local area L. Each ant can compute the local area entropy using equation 6.

$$E(L\,|\,o_i) = -\sum_{o_j \in Neigh_{s\times s}(r)} p_{i,j} \times \frac{\log_2(p_{i,j})}{\log_2\left|Neigh_{s\times s}(r)\right|} \tag{6}$$

where the probability $p_{i,j}$ indicates that we have a decisive opinion about central object o_i considering a local area object o_j in its local area L. The probability $p_{i,j}$ is obtained according to equation 7.

$$p_{i,j} = \frac{2\times\left|D(o_i, o_j)\right|}{n} \tag{7}$$

where n (n=|$Neigh_{s\times s}(r)$|) is the number of neighbors. Distance function $D(o_i,o_j)$ between each pair of objects is measured according to equation 8.

$$D(o_i, o_j) = \frac{d(o_i, o_j)}{norm(o_i)} - 0.5 \tag{8}$$

where $d(o_i,o_j)$ is Euclidian distance defined by equation 2, and $norm(o_i)$ is defined as maximum distance of object o_i with its neighbors. It is calculated according to equation 9.

$$norm(o_i) = \max_{o_j \in Neigh_{s\times s}(r)} d(o_i,o_j) \tag{9}$$

Now the function $H(L|o_i)$ is defined as equation 10.

$$H(L \mid o_i) = 1 - E(L \mid o_i) \tag{10}$$

Three examples of local area objects on a 3×3 (=9) neighborhood depicted in the Fig. 1. Different classes with different colors are displayed.

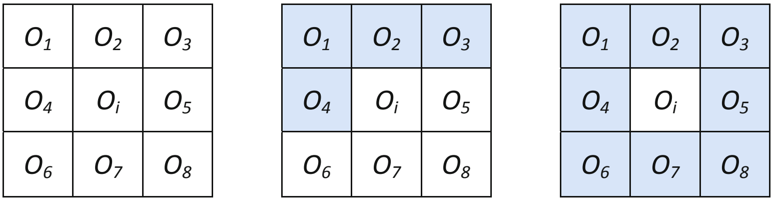


Fig. 1. Examples of local area objects

When the data objects in the local area L and central object of the local area L exactly belong to a same cluster, i.e. their distances are almost uniform and low values, such as the shape or the form depicted by the left rectangle of Fig. 1, uncertainty is low and $H(L|o_i)$ is far from one and near to 0. When the data objects in the local area L and central object of the local area L belong to some completely different separate clusters, i.e. their distances are almost uniform and high values, such as the shape or the form depicted by the right rectangle of Fig. 1, uncertainty is again low and $H(L|o_i)$ is far from one and near to 0. But in the cases of the form depicted by the middle rectangle of Fig. 1 where some data objects in the local area L and central object of the local area L exactly belong to a same cluster and some others does not, i.e. the distances are not uniform, the uncertainty is high and $H(L|o_i)$ is far from 0 and close to 1. So the function $H(L|o_i)$ can provide ants with a metric that its high value indicates the current position is a boundary area and its low value indicates the current position is not a boundary area.

In ant-based clustering, two types of pheromone are employed: (a) cluster pheromone and (b) object pheromone. Cluster pheromone guides the loaded ants to valid clusters for a possible successful dropping. Object pheromone guides the unloaded ants to lose object for a possible successful picking-up.

Each loaded ant deposits some cluster pheromone on the current position and positions of its neighbors after a successful dropping of an object to guide other ants for a place to unload their objects. The cluster pheromone intensity deposited in location j, by m ants in the colony at time t is calculated by the equation 11.

$$rc_j(t) = \sum_{a=1}^{m}\left[\mu^{(t-t_a^1)} \times C \times E(L \mid o_j)\right] \tag{11}$$

where C is cluster pheromone constant, t^1_a is the time step at that a-th cluster pheromone is deposited at position j, and μ is evaporation coefficient. On other hand, an unloaded ant deposits some object pheromone after a successful picking-up of an object to guide other agents for a place to take the objects. The object pheromone intensity deposited in location j, by m ants in the colony at time t is calculated by the equation 12.

$$ro_j(t) = \sum_{a=1}^{m}\left[\mu^{(t-t_a^2)} \times O \times H(L \mid o_j)\right] \tag{12}$$

where O is object pheromone constant, and t^2_a is the time step at that a-th object pheromone is deposited at position j. Transmission probabilities of an unloaded ant based on that ant moves from the current location i to next location j from its neighborhood can be calculated according to equation 13.

$$P_j(t) = \begin{cases} 1/w & if \sum_{j=1}^{w} ro_j(t) = 0 \forall j \in N_{dir} \\ \dfrac{ro_j(t)}{\sum_{j=1}^{n} ro_j(t)} & otherwise \end{cases} \tag{13}$$

Transmission probabilities of a loaded ant based on that ant moves from the current location i to next location j from its neighborhood can be calculated according to equation 14.

$$P_j(t) = \begin{cases} 1/w & if \sum_{j=1}^{w} rc_j(t) = 0 \forall j \in N_{dir} \\ \dfrac{rc_j(t)}{\sum_{j=1}^{n} rc_j(t)} & otherwise \end{cases} \tag{14}$$

where N_{dir} is the set of possible w actions (possible w directions to move) from current position i.

3 Proposed Ant Colony Clustering Approach

In this section the modified version of ant clustering and its new space defined is presented.

Algorithm 2. Modified ant colony clustering

```
Input:
  QD, itr, q, AntNum, Data, O, C, $k_1$, $k_2$, $v_{max}$, period, thr, st, distributions of $v, \alpha, \mu$
Initializing parameter using distributions of $v$, $\alpha$, $\mu$;
For each ant a
    Place random a in a position not occupied by other ants in a plane QD*QD;
For each object o
    Place random o in a position not occupied by other objects in the plane QD*QD;
Success (1: ant) = 0;
Failure (1: ant) = 0;
For t=1: itr
    For each ant a
        g = select a random number uniformly from range [0, 1];
        r= Position (a)
        If (loaded (a) and (is-empty (r)))
            If ($g < p_{drop}$)
                o= drop (a);
                Put (r, o);
                Save (o, r, q);
            end;
        Else if (not (loaded (a) or (is-empty (r))))
            If ($g < p_{pic}$)
                o = remove(r);
                Pick-up (a, o);
                Search&Jump (a, o);
                Success (a) = Success (a) +1;
            Else
                Failure (a) = Failure (a) +1;
            end;
        end;
        Else
            Wander (a, v, $N_{dir}$); // considering the defined pheromone
        end;
    end;
    If (t mod period == 0)
        For each ant a
            If (Success (a)/ (Failure (a) +Success (a))> thr)
                $\alpha(a) = \alpha(a) + st$;
            Else
                $\alpha(a) = \alpha(a) - st$;
            end;
        end;
    end;
end;
```

3.1 Modified Ant Colony Clustering

As mentioned before, combining the information entropy and the mean similarity as a new metric to existing models in order to detect rough areas of spatial clusters, dense clusters and troubled borders of the clusters that are wrongly merged is employed.

When the data objects in the local area *L* and central object of the local area *L* exactly belong to a same cluster, i.e. their distances are almost uniform and low values, such as the shape or the form depicted by the left rectangle of Fig. 1, uncertainty is low and $H(L|o_i)$ is far from one and near to 0. When the data objects in the local area *L* and central object of the local area *L* belong to some completely different separate clusters, i.e. their distances are almost uniform and high values, such as the shape or the form depicted by the right rectangle of Fig. 1, uncertainty is again low and $H(L|o_i)$ is far from one and near to 0. But in the cases of the form depicted by the middle rectangle of Fig. 1 where some data objects in the local area *L* and central object of the local area *L* exactly belong to a same cluster and some others does not, i.e. the distances are not uniform, the uncertainty is high and $H(L|o_i)$ is far from 0 and close to 1. So the function $H(L|o_i)$ can provide ants with a metric that its high value indicates the current position is a boundary area and its low value indicates the current position is not a boundary area.

After all the above mentioned modification, the pseudo code of ant colony clustering algorithm is presented in the Algorithm 2.

For showing an exemplary running of the modified ant colony algorithm, take a look at Fig. 2. In the Fig. 2 the final result of modified ant colony clustering algorithm over Iris dataset is presented.

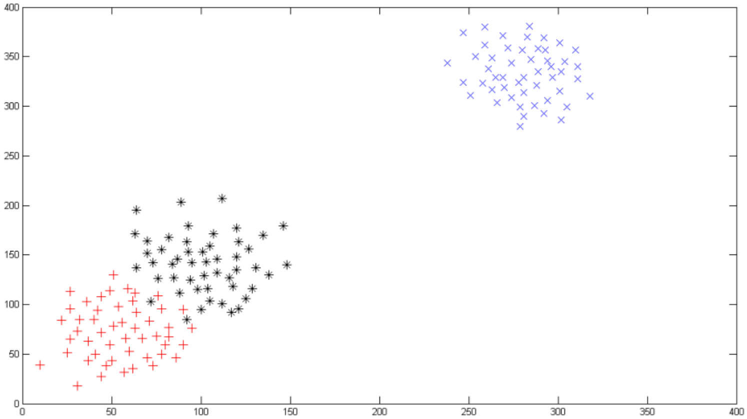


Fig. 2. Final result of modified ant colony clustering algorithm over Iris dataset

It is valuable to mention that the quantization degree parameter (*QD*), queue size parameter (*q*), ant number parameter (*AntNum*), object pheromone parameter (*O*), cluster pheromone parameter (*C*), *k1* parameter, *k2* parameter, maximum speed parameter (v_{max}), period parameter, update parameter (*thr*) evaporation parameter μ and step of update for α parameter (*st*) are respectively set to 400, 5000000, 20, 240, 1, 1, 0.1, 0.3, 150, 2000, 0.9, 0.95 and 0.01 for reaching the result of Fig. 2. Parameter α for each ant is extracted from uniform distribution of range [0.1, 1]. Parameter *v* for each ant is extracted from uniform distribution of range [1, v_{max}].

Consider that the result shown in the Fig. 2 is a well separated running of algorithm. So it is a successful running of algorithm. The algorithm may also converge to a set of overlapping clusters in an unsuccessful running.

3.2 Proposed New Space Defined by Ant Colony Algorithm

The main idea behind proposed method is using ensemble learning in the field of ant colony clustering. Due to the huge sensitiveness of modified ant colony clustering algorithm to initialization of its parameters, one can use an ensemble approach to overcome the problem of well-tuning of its parameters. The main contribution of the paper is illustrated in the Fig. 3.

As it is depicted in Fig. 3 a dataset is feed to as many as *max_run* different modified ant colony clustering algorithms with different initializations. Then we obtain *max_run* virtual 2-dimensions, one per each run modified ant colony clustering algorithm. Then by considering all these virtual 2-dimensions as new space with *2*max_run* dimensions, we reach a new data space. We can employ a clustering algorithm on the new defined data space.

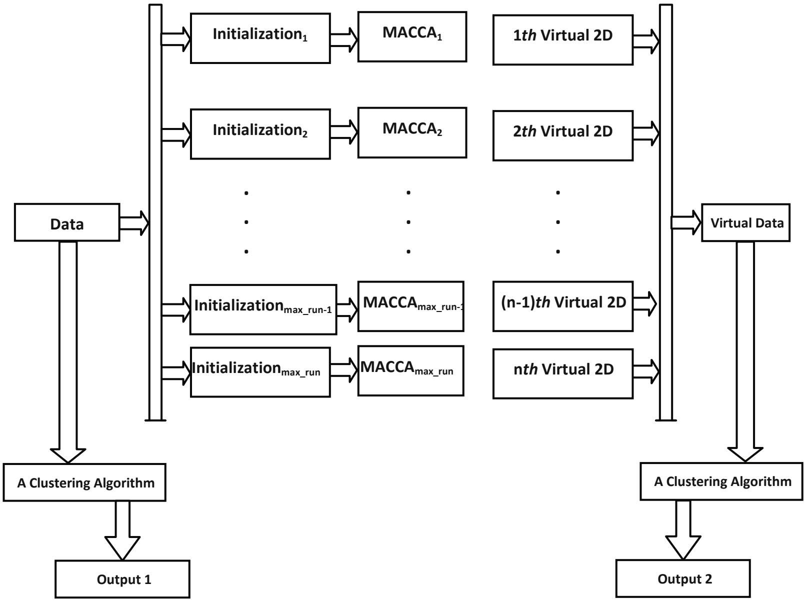


Fig. 3. Proposed framework to cluster a dataset using ant colony clustering algorithm

4 Simulation and Results

This section evaluates the result of applying proposed algorithm on some real datasets available at UCI repository (Newman et al. 1998). The main metric based on which a partition is evaluated is normalized mutual information (Strehl and Ghosh, 2002) between the output partition and real labels of the dataset is considered as the main evaluation metric of the final partition. Another alternative to evaluate a partition is the accuracy metric (Munkres, 1957). Then the settings of experimentations are given. Finally the experimental results are presented.

4.1 Experimental Settings

The quantization degree parameter (*QD*), queue size parameter (*q*), ant number parameter (*AntNum*), object pheromone parameter (*O*), cluster pheromone parameter (*C*), *k1* parameter, *k2* parameter, maximum speed parameter (v_{max}), period parameter, update parameter (*thr*) evaporation parameter μ and step of update for α parameter (*st*) are respectively set to 400, 5000000, 20, 240, 1, 1, 0.1, 0.3, 150, 2000, 0.9, 0.95 and 0.01 in all experimentations as before. Parameter α for each ant is extracted from uniform distribution of range [0.1, 1]. Parameter v for each ant is extracted from uniform distribution of range [1, v_{max}]. Fuzzy k-means (c-means) is employed as base clustering algorithm to perform final clustering over original dataset and new defined dataset. Parameter *max_run* is set to 30 in all experimentations. So the new defined space has 60 virtual features. Number of real cluster in each dataset is given to fuzzy k-means clustering algorithm in all experimentations.

Table 1. Experimental results in terms of accuracy

Dataset Name	Fuzzy k-means output 1		Fuzzy k-means output 2	
	Accuracy	Normalized Mutual Information	Accuracy	Normalized Mutual Information
Image-Segmentation	52.27	38.83	54.39	40.28
Zoo	80.08	79.09	81.12	81.24
Thyroid	83.73	50.23	87.94	59.76
Soybean	90.10	69.50	94.34	80.30
Iris	90.11	65.67	93.13	75.22
Wine	74.71	33.12	76.47	35.96

As it is inferred from the Table 1, the new defined feature space is better clustered by a base clustering algorithm rather than the original space.

4.2 Results

Table 1 shows the performance of the fuzzy clustering in both original and defined spaces in terms of accuracy and normalized mutual information. All experiments are reported over means of 10 independent runs of algorithm. It means that experimentations are done by 10 different independent runs and the final results are averaged and reported in the Table 1.

5 Conclusion

In this paper a new clustering ensemble framework is proposed which is based on a ant colony clustering algorithm and ensemble concept. In the proposed framework we use a set of modified ant colony clustering algorithms and produce a intermediate space considering their outputs totally as a defined virtual space. After producing the virtual space we employ a base clustering algorithm to obtain final partition. The experiments show that the proposed framework outperforms in comparison with the clustering over original data space. It is concluded that new defined the feature space is better clustered by a base clustering algorithm rather than the original space.

References

1. Alizadeh, H., Minaei, B., Parvin, H., Moshki, M.: An Asymmetric Criterion for Cluster Validation. In: Mehrotra, K.G., Mohan, C., Oh, J.C., Varshney, P.K., Ali, M. (eds.) Developing Concepts in Applied Intelligence. SCI, vol. 363, pp. 1–14. Springer, Heidelberg (in press, 2011)
2. Faceli, K., Marcilio, C.P., Souto, D.: Multi-objective Clustering Ensemble. In: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (2006)
3. Newman, C.B.D.J., Hettich, S., Merz, C.: UCI repository of machine learning databases (1998), http://www.ics.uci.edu/~mlearn/MLSummary.html
4. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
5. Azimi, J., Cull, P., Fern, X.: Clustering Ensembles Using Ants Algorithm. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009. LNCS, vol. 5601, pp. 295–304. Springer, Heidelberg (2009)
6. Tsang, C.H., Kwong, S.: Ant Colony Clustering and Feature Extraction for Anomaly Intrusion Detection. SCI, vol. 34, pp. 101–123 (2006)
7. Liu, B., Pan, J., McKay, R.I(B.): Incremental Clustering Based on Swarm Intelligence. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 189–196. Springer, Heidelberg (2006)
8. Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chretien, L.: The dynamics of collective sorting robot-like ants and ant-like robots. In: International Conference on Simulation of Adaptive Behavior: From Animals to Animates, pp. 356–363. MIT Press, Cambridge (1991)
9. Lumer, E.D., Faieta, B.: Diversity and adaptation in populations of clustering ants. In: International Conference on Simulation of Adaptive Behavior: From Animals to Animates, pp. 501–508. MIT Press, Cambridge (1994)
10. Munkres, J.: Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)

Global Optimization with the Gaussian Polytree EDA

Ignacio Segovia Domínguez, Arturo Hernández Aguirre,
and Enrique Villa Diharce

Center for Research in Mathematics
Guanajuato, México
{ijsegoviad,artha,villadi}@cimat.mx

Abstract. This paper introduces the Gaussian polytree estimation of distribution algorithm, a new construction method, and its application to estimation of distribution algorithms in continuous variables. The variables are assumed to be Gaussian. The construction of the tree and the edges orientation algorithm are based on information theoretic concepts such as mutual information and conditional mutual information. The proposed Gaussian polytree estimation of distribution algorithm is applied to a set of benchmark functions. The experimental results show that the approach is robust, comparisons are provided.

Keywords: Polytrees, Estimation of Distribution Algorithm, Optimization.

1 Introduction

The polytree ia a graphical model with wide applications in artificial intelligence. For instance, in belief networks the polytrees are the de-facto graph because they support probabilistic inference in linear time [13]. Other applications make use of polytrees in a rather similar way, that is, polytrees are frequently used to model the joint probability distribution (JPD) of some data. Such JPD is also called a factorized distribution because the tree encodes a joint probability as a product of conditional distributions.

In this paper we are concerned with the use of polytrees and their construction and simulation algorithms. Further more, we asses the improvement that polytrees bring to the performance of Estimation of Distribution Algorithms (EDAs). As mentioned the polytree graphs have been applied by J. Pearl to belief networks [13], but also Acid and de Campos researched them in causal networks [1], [14]. More recently, M. Soto applied polytrees to model distributions in EDAs and came up with the polytree approximation distribution algorithm, known as PADA [11]. However, note that in all the mentioned approaches the variables are binary. The goal of this paper is to introduce the polytree for continuous variables, that is, a polytree in continuous domain with Gaussian variables and its application to EDAs for optimization. The proposed approach is called the Gaussian Polytree EDA. Polytrees with continuous variables have been studied

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 165–176, 2011.

by Ouerd [12], [9]. In this paper we extend a poster presented [16] and we further develop the work of Ouerd [12]. We introduce two new algorithmic features to the gaussian polytree: 1) a new orientation principle based on conditional mutual information. We also prove that our approach is correct, 2) overfitting control of the model through a comparison of conditional and marginal mutual information strengths. The determination of the threshold value is also explained.

This paper is organized as follows. Section 2 describes two polytree algorithms in discrete variables; Section 3 explains how to build a Gaussian polytree while Section 4 provides the implementation details. Section 5 describes two sets of experiments and provides a comparison with other related approaches. Section 6 provides the conclusions and lines of future research.

2 Related Work

A polytree is a directed acyclic graph (DAG) with no loops when the edges are undirected (only one path between any two nodes) [6],[8]. For binary variables the polytree approximation distribution algorithm (PADA) is the first work to propose the use of polytrees in estimation distribution algorithm [11]. The construction algorithm of PADA uses (marginal) mutual information and conditional mutual information as a measure of the dependency. Thus, a node X_k is made head to head whenever the conditional mutual information $CMI(X_i, X_j|X_k)$ is greater than the marginal mutual information $MI(X_i, X_j)$). Thus, the head to head node means that the information shared by two nodes X_i, X_j increases when the third node X_k is included. For overfitting control two parameters ϵ_1, ϵ_2 aim to filter out the (weak) dependencies. However no recommendations about how to set these parameters is given in the PADA literature.

A Gaussian polytree is a factorized representation of a multivariate normal distribution [10],[4]. Its JPDF is a product of Gaussian conditional probabilites times the product of the probabilities of the root nodes (R), as follows: $JPDF(X_1, X_2, \ldots, X_n) = \prod_{\forall i \in R} P(X_i) \prod_{\forall j \notin R} P(X_j|pa(X_j))$. A recent approach uses a depth first search algorithm for edge orientation [9]. Based on the previous work of Rebane and Pearl [15],[13], Ouerd at al. assume that a Chow & Liu algorithm is ran to deliver a dependence tree from the data [9]. Then they propose to orient the edges by traversing the dependence tree in a depth first search order. Articulation points and causal basins must be detected first. With their approach they try to solve four issues (not completely solved by Rebane and Pearl) such as how to traverse the tree, and what to do with the edges already traversed. For edge orientation their algorithm performs a marginal independence test on the parents X and Y of a node Z to decide if Z has X and Y as parents. If they are independent the node Z is a head to head node.

3 Building the Gaussian Polytree

In the following we describe the main steps needed to construct a Gaussian polytree.

1. The Gaussian Chow & Liu tree. The first step to construct a Gaussian polytree is to construct a *Gaussian Chow & Liu dependence tree* (we use the same approach of the binary dependence tree of Chow & Liu [3]). Recall *mutual information* is the measure to estimate dependencies in Chow & Liu algorithm. The algorithm randomly chooses a node and declares it the root. Then the Kruskal algorithm is used to create a maximum weight spanning tree. The tree thus created maximizes the total mutual information, and it is the best approximation to the true distribution of the data whenever that distribution comes from a tree like factorization. A Gaussian Chow & Liu tree is created in a way similar to the discrete variables case. Mutual information is also the maximum likelihood estimator, and whenever a multivariate normal distribution is factorized as the product of second order distributions the Gaussian Chow & Liu tree is the best approximation. For normal variables, mutual information is defined as:

$$MI(X,Y) = -\frac{1}{2}\log\left(1 - r_{x,y}^2\right). \tag{1}$$

   The term $r_{x,y}$ is the Pearson's correlation coefficient which for Gaussian variables is defined as:

$$r_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} \tag{2}$$

2. Edge orientation. The procedure to orient the edges of the tree is based on the orienting principle [15]: if in a triplet $X - Z - Y$ the variables X and Y are independent then Z is a head to head node with X and Y as parents, as follows: $X \rightarrow Z \leftarrow Y$. Similarly, if in a triplet $X \rightarrow Z - Y$ the variables X and Y are independent then Z is a head to head node with X and Y as parents:$X \rightarrow Z \leftarrow Y$; otherwise Z is the parent of Y: $X \rightarrow Z \rightarrow Y$.
   In this paper we propose information theoretic measures such a conditional mutual information (CMI) and (marginal) mutual information (MI) to estimate the dependency between variables.
   Proposed orientation based on information measures: for any triplet $X - Z - Y$, if $CMI(X,Y|Z) > MI(X,Y)$ then Z is a head to head node with X and Y as parents, as follows: $X \rightarrow Z \leftarrow Y$.
   Proof. We shall prove that the proposed measure based on mutual information finds the correct orientation. That is, (in Figure 1 the four possible models made with three variables are shown), model M_4, head to head, is the correct one for $CMI(X,Y|Z) > MI(X,Y)$.
   The quality of the causal models shown in the Figure 1 can be expressed by its log-likelihood. If the parents of any node X_i is the set of nodes $pa(X_i)$, the negative of the log-likelihood of a model M is [5]:

$$-\,ll(M) = \sum_{i=1}^{n} H(X_i|pa(X_i)) \tag{3}$$

   where $H(X_i|pa(X_i))$ is the conditional entropy of X_i given its parents $pa(X_i)$. It is well known that the causal models M_1, M_2 and M_3 are equivalent,

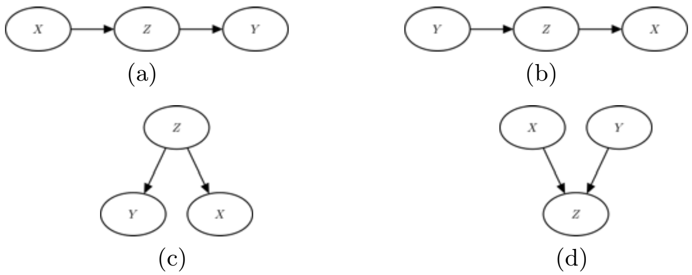


Fig. 1. The causal models that can be obtained with three variables X, Y y Z. (a) Model M_1. (b) Model M_2. (c) Model M_3. (d) Model M_4.

or indistinguishable in probability [15]. The negative log-likelihood are the Equations 4, 5 and 6, respectively.

$$\begin{aligned} -ll(M_1) &= H(X) + H(Z|X) + H(Y|Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) \\ &\quad -H(X,Y,Z) + H(X,Y,Z) \\ &= H(X,Y,Z) + CMI(X,Y|Z) \end{aligned} \tag{4}$$

$$\begin{aligned} -ll(M_2) &= H(Z) + H(X|Z) + H(Y|Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) \\ &\quad +H(X,Y,Z) - H(X,Y,Z) \\ &= H(X,Y,Z) + CMI(X,Y|Z) \end{aligned} \tag{5}$$

$$\begin{aligned} -ll(M_3) &= H(Y) + H(Z|Y) + H(X|Z) \\ &= H(X,Z) + H(Y,Z) - H(Z) \\ &\quad -H(X,Y,Z) + H(X,Y,Z) \\ &= H(X,Y,Z) + CMI(X,Y|Z) \end{aligned} \tag{6}$$

For the head to head model (M_4), the negative of the log-likelihood is Equation 7.

$$\begin{aligned} -ll(M_4) &= H(X) + H(Y) + H(Z|X,Y) \\ &= H(X) + H(Y) + H(X,Y,Z) - H(X,Y) \\ &= H(X,Y,Z) + MI(X,Y) \end{aligned} \tag{7}$$

The best model is that one with the smallest negative log-likelihood or smallest summation of conditional entropy. When is the negative log-likelihood of Model M_4 smaller than the log-likelihood of model M_1 or M_2 or M_3 ?

$$H(X,Y,Z) + MI(X,Y) < H(X,Y,Z) + CMI(X,Y|Z) \tag{8}$$

The answer is in Equation 8. When the conditional mutual information $CMI(X,Y|Z)$ is larger than $MI(X,Y)$ the model M_4 has smaller negative log-likelihood value, therefore, $M4$ is the "best". □

In this work, the edge orientation principle runs on the depth first search algorithm [9]. The principle is applied to every pair of parent nodes in the

following way. Assume node A has nodes B, C, and D as candidate parents. There are 3 triplets to test: $B-A-C$, $B-A-D$ and $C-A-D$. As soon as a pair agrees with the proposed orientation principle, the edges are oriented as a head to head node. When the next triplet is tested but one of the edges is already directed the new test do not modify its direction.
The equation to compute the conditional mutual information of Gaussian variables is:

$$CMI(X,Y|Z) = \frac{1}{2}\log\left[\frac{\sigma_x^2\sigma_y^2\sigma_z^2\left(1-r_{xz}^2\right)\left(1-r_{yz}^2\right)}{|\Sigma_{xyz}|}\right] \tag{9}$$

3. Over-fitting control. The inequality
$MI(X,Y) < CMI(X,Y|Z)$ could be made true due to the small biases of the data and creating false positive parents. As a rule, the larger the allowed number of parents the larger the over-fitting. Multi parent nodes are great for polytrees but these nodes and their parents must be carefully chosen. A hypothesis test based on a non parametric bootstrap test over the data vectors X, Y and Z can be performed to solve the over-fitting problem. In this approach we used the statistic $\hat{\theta} = \overline{CMI(X^*,Y^*|Z^*)} - \overline{MI(X^*,Y^*)}$, the significance level 5%, null hypothesis $H_0 = \overline{CMI(X^*,Y^*|Z^*)} \leq \overline{MI(X^*,Y^*)}$ and alternative hypothesis $H_1 = \overline{CMI(X^*,Y^*|Z^*)} > \overline{MI(X^*,Y^*)}$. However this approach is computationally expensive. A better approach would be based on a threshold value but which value? Hence the question is: how many times the CMI must be larger than the MI as to represent true parents? Which is a good threshold value?. Empirically we solve this question by randomly creating a huge database of triplet-vectors X, Y and Z (from random gaussian distributions) that made true the inequality $MI(X,Y) < CMI(X,Y|Z)$. Within this large set there are two subsets: triplets that satisfy the null hypothesis and those that not. We found out that false parents are created in 95% of the cases when $\frac{CMI(X,Y|Z)}{MI(X,Y)} < 3$. Therefore the sought threshold value is 3. Thus a head to head node is created whenever $\frac{CMI(X,Y|Z)}{MI(X,Y)} \geq 3$.

4 Aspects of the Gaussian Polytree EDA

In the previous section we explained the algorithm to build a gaussian polytree. An Estimation Distribution Algorithm was created using our model. Two aspects of the Gaussian polytree EDA are important to mention.

1. Data simulation. The procedure to obtain a new population (or new samples) from a polytree follows the common strategy of sampling from conditional Gaussian variables. If variable X_i is conditioned on $Y = \{X_j, X_k, \ldots, X_z\}$, with $X_i \notin Y$, their conditional Gaussian distribution:

$$\mathcal{N}_{X_i|Y=\mathbf{y}}\left(\mu_{X_i|Y=\mathbf{y}}, \Sigma_{X_i|Y=\mathbf{y}}\right)$$

can be simulated using the conditional mean

$$\mu_{X_i|Y=\mathbf{y}} = \mu_{X_i} + \Sigma_{X_iY}\Sigma_{YY}^{-1}\left(\mathbf{y}-\mu_Y\right) \tag{10}$$

and the conditional covariance:

$$\varSigma_{X_i|Y=\mathbf{y}} = \varSigma_{X_iX_i} - \varSigma_{X_iY}\varSigma_{YY}^{-1}\varSigma_{YX_i} \tag{11}$$

The simulation of samples at time t follows the gaussian polytree structure. If X_i^t has no parents then $X_i^t \sim \mathcal{N}(\mu_{X_i^{t-1}}, \varSigma_{X_i^{t-1}})$; otherwise X_i^t follow the gaussian distribution conditioned to $Y = \mathbf{y}^{t-1}$. This method adds exploration to the gaussian polytree EDA. Notice it is different of common ancestral sampling.

2. Selection. In EDAs truncation selection is commonly used. Our approach differs. We select the K best individuals whose fitness is better than the average fitness of the entire population. By including all members of the population the average gets a poor value. Then the selection pressure is low and many different individuals (high diversity) are selected and used as information to create the next polytree.

5 Experiments

The Gaussian polytree EDA is tested in two sets of benchmark functions.

5.1 Experiment 1: Convex Functions

This set of 9 convex functions was solved using the IDEA algorithm adapted with mechanisms to avoid premature convergence and to improve the convergence speed [7],[2]. The functions are listed in Table 3. In [7] the mechanism increases or decreases the variance accordingly to the rate the fitness function improves. In [2] the mechanism computes the shift of the mean in the direction of the best individual in the population. These mechanism are necessary due to premature convergence of the IDEA algorithm. Notice that the Gaussian polytree EDA does not need any additional mechanism to converge to the optimum. 30 runs were made for each problem.

Initialization. Asymmetric initialization is used for all the variables: $X_i \in [-10, 5]$.
Population size. For a problem in l dimensions, the population is $2 \times (10(l^{0.7}) + 10)$ [2]
Stopping conditions. Maximum number of fitness function evaluations is reached: 1.5×10^5; or target error smaller than 1×10^{-10}; or no improving larger than 1×10^{-13} is detected after 30 generations and the mean of l standard deviations, one for each dimension, is less than 1×10^{-13}.

The Figure 2 shows the best number of evaluations needed to reach the target error for dimensions 2, 4, 8, 10, 20, 40, and 80. The success rate VS the problem dimensionality is listed in Table 1 and Table 2 details the number of evaluations found in our experiments.

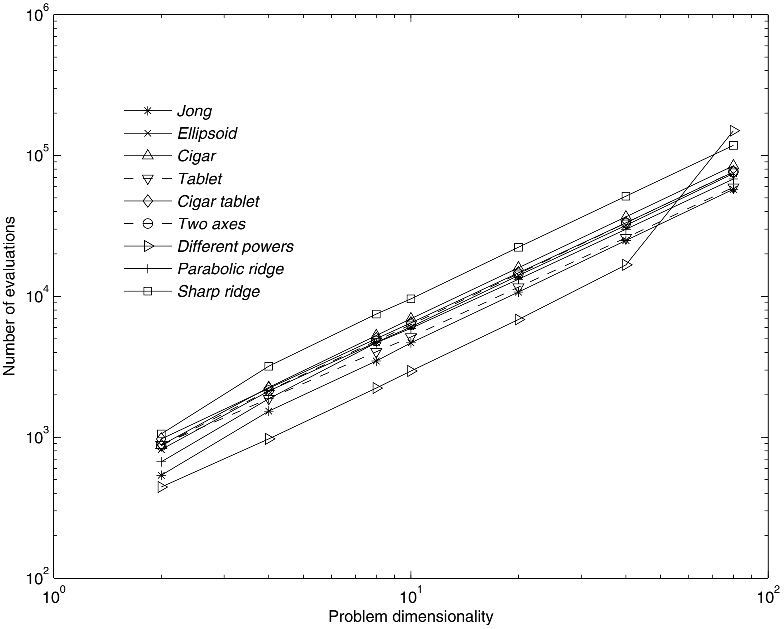


Fig. 2. Best number of evaluations VS problem dimensionality

Comments to Experiment 1. Note that the increment in the number of evaluations increases proportional to the increment in the dimensionality of the problem. The gaussian polytree EDA maintains a high success rate of global convergence, even in dimension 80. Out of these functions, just the different powers function (and slightly the two axes) were difficult to solve.

5.2 Experiment 2: Non-convex Functions

In this experiment we use four functions that Larrañaga and Lozano tested with different algorithms, including the estimation of Gaussian network algorithm

Table 1. Success rate of functions (%) VS problem dimensionality

Function	2-D	4-D	8-D	10-D	20-D	40-D	80-D
$\mathcal{F}_1$	100	100	100	100	100	100	100
$\mathcal{F}_2$	96.6	96.6	93.3	90.0	96.6	90.0	86.6
$\mathcal{F}_3$	96.6	93.3	86.6	86.6	93.3	96.6	93.3
$\mathcal{F}_4$	100	90.0	96.6	100	100	100	100
$\mathcal{F}_5$	90.0	93.3	93.3	100	96.6	100	100
$\mathcal{F}_6$	96.6	90.0	83.3	80.0	63.3	70.0	60.0
$\mathcal{F}_7$	100	100	96.6	93.3	73.3	26.6	0.0
$\mathcal{F}_8$	80.0	73.3	83.3	86.6	83.3	90.0	100
$\mathcal{F}_9$	73.3	83.3	96.6	100	100	100	100

Table 2. Number of evaluations performed by the Gaussian polytree EDA needed to reach the target error in 30 repetitions (see stopping conditions)

$\mathcal{F}_i$	Dim	Best	Worst	Mean	Median	SD
$\mathcal{F}_1$	2	5.3700 e2	8.2500 e2	7.3300 e2	7.5200 e2	6.2433 e1
	4	1.5340 e3	1.8090 e3	1.6739 e3	1.6770 e3	5.9753 e1
	8	3.4780 e3	3.9450 e3	3.7791 e3	3.7980 e3	9.5507 e1
	10	4.6760 e3	5.1220 e3	4.8663 e3	4.8690 e3	9.2939 e1
	20	1.0744 e4	1.1258 e4	1.1048 e4	1.1069 e4	1.3572 e2
	40	2.4931 e4	2.5633 e4	2.5339 e4	2.5308 e4	1.8670 e2
	80	5.7648 e4	5.8966 e4	5.8510 e4	5.8574 e4	3.1304 e2
$\mathcal{F}_2$	2	8.1800 e2	3.2950 e3	1.0650 e3	1.0115 e3	4.2690 e2
	4	2.1280 e3	5.8800 e3	2.3583 e3	2.2495 e3	6.6716 e2
	8	4.7180 e3	1.0001 e5	8.2475 e3	4.8910 e3	1.7363 e4
	10	6.0830 e3	2.0292 e4	7.2357 e3	6.3480 e3	2.9821 e3
	20	1.4060 e4	2.4260 e4	1.4686 e4	1.4303 e4	1.8168 e3
	40	3.1937 e4	5.1330 e4	3.4468 e4	3.2749 e4	5.4221 e3
	80	7.4495 e4	1.2342 e5	8.0549 e4	7.5737 e4	1.2893 e4
$\mathcal{F}_3$	2	8.8000 e2	3.5210 e3	1.0819 e3	1.0145 e3	4.6461 e2
	4	2.2600 e3	7.2280 e3	2.6692 e3	2.4375 e3	9.8107 e2
	8	5.2700 e3	1.5503 e4	6.3378 e3	5.5220 e3	2.3176 e3
	10	6.9430 e3	1.3732 e4	7.9081 e3	7.1060 e3	2.0858 e3
	20	1.5956 e4	2.6813 e4	1.6900 e4	1.6237 e4	2.6287 e3
	40	3.6713 e4	5.4062 e4	3.7592 e4	3.7017 e4	3.1153 e3
	80	8.4462 e4	1.1823 e5	8.7323 e4	8.5144 e4	8.2764 e3
$\mathcal{F}_4$	2	8.8300 e2	1.1120 e3	9.9520 e2	9.8900 e2	5.8534 e1
	4	1.8830 e3	5.8250 e3	2.3616 e3	1.9990 e3	1.1030 e3
	8	4.0430 e3	9.3870 e3	4.4143 e3	4.2545 e3	9.4333 e2
	10	5.1480 e3	5.6070 e3	5.4052 e3	5.4285 e3	1.1774 e2
	20	1.1633 e4	1.2127 e4	1.1863 e4	1.1861 e4	1.0308 e2
	40	2.6059 e4	2.6875 e4	2.6511 e4	2.6487 e4	2.2269 e2
	80	5.9547 e4	6.1064 e4	6.0308 e4	6.0302 e4	3.6957 e2
$\mathcal{F}_5$	2	9.7300 e2	3.6130 e3	1.3396 e3	1.1155 e3	7.4687 e2
	4	2.2230 e3	6.0680 e3	2.6141 e3	2.3760 e3	9.0729 e2
	8	5.0060 e3	1.0809 e4	5.5754 e3	5.2045 e3	1.4230 e3
	10	6.4820 e3	6.9730 e3	6.7031 e3	6.7075 e3	1.1929 e2
	20	1.4687 e4	2.7779 e4	1.5381 e4	1.4983 e4	2.3449 e3
	40	3.3287 e4	3.4203 e4	3.3852 e4	3.3865 e4	2.0564 e2
	80	7.6250 e4	7.8009 e4	7.7247 e4	7.7359 e4	3.8967 e2
$\mathcal{F}_6$	2	8.7100 e2	2.9510 e3	1.0655 e3	9.9550 e2	3.5942 e2
	4	2.1480 e3	5.5960 e3	2.5739 e3	2.2475 e3	1.0015 e3
	8	4.8380 e3	1.6298 e4	6.0937 e3	5.0160 e3	2.6565 e3
	10	6.3130 e3	2.3031 e4	8.1936 e3	6.5415 e3	3.8264 e3
	20	1.4455 e4	6.0814 e4	2.0558 e4	1.4919 e4	1.0252 e4
	40	3.3222 e4	6.2568 e4	3.9546 e4	3.3955 e4	9.2253 e3
	80	7.6668 e4	1.0019 e5	8.6593 e4	7.8060 e4	1.1221 e4
$\mathcal{F}_7$	2	4.4400 e2	6.2100 e2	5.2970 e2	5.3450 e2	5.1867 e1
	4	9.7500 e2	1.2580 e3	1.1103 e3	1.1100 e3	6.8305 e1
	8	2.2360 e3	7.3335 e4	4.7502 e3	2.4010 e3	1.2953 e4
	10	2.9530 e3	9.9095 e4	7.7189 e3	3.1475 e3	1.8871 e4
	20	6.8480 e3	1.0011 e5	3.1933 e4	7.2465 e3	4.1782 e4
	40	1.6741 e4	1.0017 e5	7.7923 e4	1.0003 e5	3.7343 e4
	80	1.5001 e5	1.5024 e5	1.5010 e5	1.5008 e5	7.1759 e1
$\mathcal{F}_8$	2	6.7000 e2	3.8730 e3	1.3424 e3	8.5950 e2	1.0699 e3
	4	1.8780 e3	8.8220 e3	3.2186 e3	2.2065 e3	1.8858 e3
	8	4.6880 e3	1.0773 e4	5.7467 e3	4.8275 e3	2.1246 e3
	10	5.9350 e3	1.2863 e4	7.0149 e3	6.1555 e3	2.2485 e3
	20	1.3228 e4	2.6804 e4	1.5504 e4	1.3667 e4	4.3446 e3
	40	2.9959 e4	8.3911 e4	3.3521 e4	3.0451 e4	1.0781 e4
	80	6.8077 e4	7.0542 e4	6.9092 e4	6.9069 e4	4.7975 e2
$\mathcal{F}_9$	2	1.0560 e3	4.2000 e3	2.0126 e3	1.3910 e3	1.1536 e3
	4	3.1980 e3	7.5810 e3	4.0188 e3	3.4055 e3	1.4445 e3
	8	7.4930 e3	1.4390 e4	7.9337 e3	7.7140 e3	1.2243 e3
	10	9.6110 e3	1.0325 e4	1.0013 e4	9.9930 e3	1.5436 e2
	20	2.2342 e4	2.3122 e4	2.2776 e4	2.2780 e4	1.9712 e2
	40	5.1413 e4	5.2488 e4	5.1852 e4	5.1827 e4	2.4254 e2
	80	1.1796 e5	1.2033 e5	1.1896 e5	1.1904 e5	5.3493 e2

Table 3. Set of convex functions of Experiment 1

Name	Alias	Definition
Sphere	$\mathcal{F}_1$	$\sum_{i=1}^{N} X_i^2$
Ellipsoid	$\mathcal{F}_2$	$\sum_{i=1}^{N} 10^{6\frac{i-1}{N-1}} X_i^2$
Cigar	$\mathcal{F}_3$	$X_1^2 + \sum_{i=2}^{N} 10^6 X_i^2$
Tablet	$\mathcal{F}_4$	$10^6 X_1^2 + \sum_{i=2}^{N} X_i^2$
Cigar Tablet	$\mathcal{F}_4$	$X_1^2 + \sum_{i=2}^{N-1} 10^4 X_i^2 + 10^8 X_N^2$
Two Axes	$\mathcal{F}_6$	$\sum_{i=1}^{[N/2]} 10^6 X_i^2 + \sum_{i=[N/2]}^{N} X_i^2$
Different Powers	$\mathcal{F}_7$	$\sum_{i=1}^{N} \lvert X_i \rvert^{2+10\frac{i-1}{N-i}}$
Parabolic Ridge	$\mathcal{F}_8$	$-X_1 + 100 \sum_{i=2}^{N} X_i^2$
Sharp Ridge	$\mathcal{F}_9$	$-X_1 + 100\sqrt{\sum_{i=2}^{N} X_i^2}$

($EGNA$). $EGNA$ is interesting for this comparison because it is a graph with continuous variables built with scoring metrics such as the Bayesian information criteria (BIC). The precision matrix is created from the graph structure which allows none or more parents to any node. Therefore, the Gaussian polytree and the $EGNA$ allow several parents.
The experimental settings are the following:

Population size. For a problem in l dimensions, the population is $2 \times (10(l^{0.7}) + 10)$ [2]
Stopping conditions. Maximum number of fitness function evaluations is: 3×10^5; or target error smaller than 1×10^{-6}, 30 repetitions. Also stop when no improving larger than 1×10^{-13} is detected after 30 generations and the mean of l standard deviations, one for each dimension, is less than 1×10^{-13}.

The set of test functions is shown in Table 4. Experiments were performed for dimensions 10 and 50. The comparison for the Sphere function is shown in Figure 5, for the Rosenbrock function in Table 6, for the Griewangk in Table 7, and for the Ackley function in Table 8.

Table 4. Set of test functions of Experiment 2

Name	Alias	Definition	Domain
Sphere	$\mathcal{F}_1$	$\sum_{i=1}^{N} X_i^2$	$-600 \leq X_i \leq 600$
Rosenbrock	$\mathcal{F}_2$	$\sum_{i=1}^{N-1} \left[(1 - X_i)^2 + 100\left(X_{i+1} - X_i^2\right)^2\right]$	$-10 \leq X_i \leq 10$
Griewangk	$\mathcal{F}_4$	$\sum_{i=1}^{N} \frac{X_i^2}{4000} - \prod_{i=1}^{N} \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	$-600 \leq X_i \leq 600$
Ackley	$\mathcal{F}_5$	$-20 \exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} X_i^2}\right)$	$-10 \leq X_i \leq 10$
		$- \exp\left(\frac{1}{N}\sum_{i=1}^{N} \cos(2\pi X_i)\right) + 20 + e$	

Table 5. Comparative for the Sphere function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	2.5913e-5 ± 3.71e-5	77162.4 ± 6335.4
	$EGNA_{BGe}$	7.1938e-6 ± 1.78e-6	74763.6 ± 1032.2
	$EGNA_{ee}$	7.3713e-6 ± 1.98e-6	73964 ± 1632.1
	PolyG	7.6198e-7 ± 1.75e-7	4723.9 ± 78.7
50	$EGNA_{BIC}$	1.2126e-3 ± 7.69e-4	263869 ± 29977.5
	$EGNA_{BGe}$	8.7097e-6 ± 1.30e-6	204298.8 ± 1264.2
	$EGNA_{ee}$	8.3450e-6 ± 1.04e-6	209496.2 ± 1576.8
	PolyG	8.9297e-7 ± 8.05e-8	32258.4 ± 274.1

Table 6. Comparative for the Rosenbrock function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	8.8217 ± 0.16	268066.9 ± 69557.3
	$EGNA_{BGe}$	8.6807 ± 5.87e-2	164518.7 ± 24374.5
	$EGNA_{ee}$	8.7366 ± 2.23e-2	301850 ± 0.0
	PolyG	7.9859 ± 2.48e-1	18931.8 ± 3047.6
50	$EGNA_{BIC}$	50.4995 ± 2.30	301850 ± 0.0
	$EGNA_{BGe}$	48.8234 ± 0.118	301850 ± 0.0
	$EGNA_{ee}$	48.8893 ± 1.11e-2	301850 ± 0.0
	PolyG	47.6 ± 1.52e-1	81692.2 ± 6704.7

Table 7. Comparative for the Griewangk function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	3.9271e-2 ± 2.43e-2	301850 ± 0.0
	$EGNA_{BGe}$	7.6389e-2 ± 2.93e-2	301850 ± 0.0
	$EGNA_{ee}$	5.6840e-2 ± 3.82e-2	301850 ± 0.0
	PolyG	3.6697e-3 ± 6.52e-3	60574.3 ± 75918.5
50	$EGNA_{BIC}$	1.7075e-4 ± 6.78e-5	250475 ± 18658.5
	$EGNA_{BGe}$	8.6503e-6 ± 7.71e-7	173514.2 ± 1264.3
	$EGNA_{ee}$	9.1834e-6 ± 5.91e-7	175313.3 ± 965.6
	PolyG	8.9551e-7 ± 6.24e-8	28249.8 ± 227.4

Comments to Experiment 2. The proposed Gaussian polytree EDA reaches better values than the $EGNA$ requiring lesser number of function evaluations in all function (except for the Rosenbrock were both show a similar performance).

Table 8. Comparative for the Ackley function with a dimension of 10 and 50 (optimum fitness value = 0)

Dimension	Algorithm	Best	Evaluations
10	$EGNA_{BIC}$	5.2294 ± 4.49	229086.4 ± 81778.4
	$EGNA_{BGe}$	7.9046e-6 ± 1.39e-6	113944 ± 1632.2
	$EGNA_{ee}$	74998e-6 ± 1.72e-6	118541.7 ± 2317.8
	$PolyG$	8.3643e-7 ± 1.24e-7	5551.5 ± 104.0
50	$EGNA_{BIC}$	19702e-2 ± 7.50e-3	288256.8 ± 29209.4
	$EGNA_{BGe}$	8.6503e-6 ± 3.79e-7	282059.9 ± 632.1
	$EGNA_{ee}$	6.8198 ± 0.27	301850 ± 0.0
	$PolyG$	9.4425e-7 ± 4.27e-8	36672.9 ± 241.0

6 Conclusions

In this paper we described a new EDA based on Gaussian polytrees. A polytree is a rich modeling structure that can be built with moderate computing costs. At the same time the Gaussian polytree is found to have a good performance on the tested functions. Other algorithms have shown convergence problems on convex functions and need special adaptations that the Gaussian polytree did not need. The new sampling method favors diversity of the population since it is based on the covariance matrix of the parent nodes and the children nodes. Also the proposed selection strategy applies low selection pressure to the individuals therefore improving diversity and delaying convergence.

References

1. Acid, S., de Campos, L.M.: Approximations of Causal Networks by Polytrees: An Empirical Study. In: Bouchon-Meunier, B., Yager, R.R., Zadeh, L.A. (eds.) IPMU 1994. LNCS, vol. 945, pp. 149–158. Springer, Heidelberg (1995)
2. Bosman, P.A.N., Grahl, J., Thierens, D.: Enhancing the performance of maximum-likelihood gaussian edas using anticipated mean shift. In: Proceedings of BNAIC 2008, the Twentieth Belgian-Dutch Artificial Intelligence Conference, pp. 285–286. BNVKI (2008)
3. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory IT-14(3), 462–467 (1968)
4. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
5. Dasgupta, S.: Learning polytrees. In: Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI 1999), pp. 134–141. Morgan Kaufmann, San Francisco (1999)
6. Edwards, D.: Introduction to Graphical Modelling. Springer, Berlin (1995)
7. Grahl, P.A.B.J., Rothlauf, F.: The correlation-triggered adaptive variance scaling idea. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO 2006, pp. 397–404. ACM (2006)
8. Lauritzen, S.L.: Graphical models. Clarendon Press (1996)

9. Ouerd, B.J.O.M., Matwin, S.: A formal approach to using data distributions for building causal polytree structures. Information Sciences, an International Journal 168, 111–132 (2004)
10. Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall series in Artificial Intelligence (2004)
11. Ortiz, M.S.: Un estudio sobre los Algoritmos Evolutivos con Estimacion de Distribuciones basados en poliarboles y su costo de evaluacion. PhD thesis, Instituto de Cibernetica, Matematica y Fisica, La Habana, Cuba (2003)
12. Ouerd, M.: Learning in Belief Networks and its Application to Distributed Databases. PhD thesis, University of Ottawa, Ottawa, Ontario, Canada (2000)
13. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
14. de Campos, L.M., Moteos, J., Molina, R.: Using bayesian algorithms for learning causal networks in classification problems. In: Proceedings of the Fourth International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), pp. 395–398 (1993)
15. Rebane, G., Pearl, J.: The recovery of causal poly-trees from statistical data. In: Proceedings, 3rd Workshop on Uncertainty in AI, Seattle, WA, pp. 222–228 (1987)
16. Segovia-Dominguez Ignacio, H.-A.A., Enrique, V.-D.: The gaussian polytree eda for global optimization. In: Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO 2011, pp. 69–70. ACM, New York (2011)

Comparative Study of BSO and GA for the Optimizing Energy in Ambient Intelligence

Wendoly J. Gpe. Romero-Rodríguez, Victor Manuel Zamudio Rodríguez,
Rosario Baltazar Flores, Marco Aurelio Sotelo-Figueroa,
and Jorge Alberto Soria Alcaraz

Division of Research and Postgraduate Studies, Leon Insitute of Technology,
Av. Tecnológico S/N Fracc. Ind. Julián de Obregón. C.P. 37290 León, México
{wendolyjgrr,masotelof,ryoga_kaji}@gmail.com
vic.zamudio@ieee.org, charobalmx@yahoo.com.mx

Abstract. One of the concerns of humanity today is developing strategies for saving energy, because we need to reduce energetic costs and promote economical, political and environmental sustainability. As we have mentioned before, in recent times one of the main priorities is energy management. The goal in this project is to develop a system that will be able to find optimal configurations in energy savings through management light. In this paper a comparison between Genetic Algorithms (GA) and Bee Swarm Optimization (BSO) is made. These two strategies are focus on lights management, as the main scenario, and taking into account the activity of the users, size of area, quantity of lights, and power. It was found that the GA provides an optimal configuration (according to the user's needs), and this result was consistent with Wilcoxon's Test.

Keywords: Ambient Intelligence, Energy Management, GA, BSO.

1 Introduction

The concept of Ambient Intelligence [1] presents a futuristic vision of the world emphasizing efficiency and supporting services delivered to the user, user empowerment and ease of human interaction with the environment. Nowadays one of the main concerns of humanity is energy saving strategies to reduce costs and promote environmental sustainability, taking into account that one of the objectives of ambient intelligence is to achieve control of the environment surrounding a user. In this sense, AmI technology must be designed for users to be the center of the development, rather than expecting the users to adapt to technology (ISTAG) [2]. For the case of power management in AmI we focus on light management, taking into account that it will be different according to the different activities that can be done. There is a need for a system to be able to find optimal energy configurations. In this

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 177–188, 2011.

research we are interested on finding energy efficiency under different setups, using heuristic techniques to target the environment is able to optimize the energy parameter in the ambient intelligent about stage lighting and activity to be performed by the same. Some strategies that have been used to control the illumination are Fuzzy Logic that improves energy efficiency in a lighting system with passive optical fiber, wherein the intensity measurements and occupation of a room are used by the fuzzy system to control the system [3]. Other approaches are based on collections of software agents that monitor and control a small office building using electrical devices [4]. HOMEBOTS are intelligent agent in charge of power management [5]. In our work we optimize lighting management and energy efficiency taking into account the activity to be performed, power lights, using genetic algorithms [6] and Bee Swarm Optimization Binary. Additionally, a comparison between these algorithms is performed.

2 Algorithms

2.1 Binary Bee Swarm Optimization

BSO is a combination of the algorithms PSO (Particle Swarm Optimization) [7] and Bee Algorithm (BA) [8], and uses the approach known as "social metaphor" [9]. Additionally, uses the intelligent behavior of bees for foraging, seeking better food sources along a radius search. BSO is based on taking the best of each metaheuristic to obtain the best results [10]. For updating the velocity and position of each particle Equation 1 is used, and also applies the social metaphor [11]. Probabilistically speed upgrade can translate into a new position, such as new solutions, using the sigmoid function in equation 2.

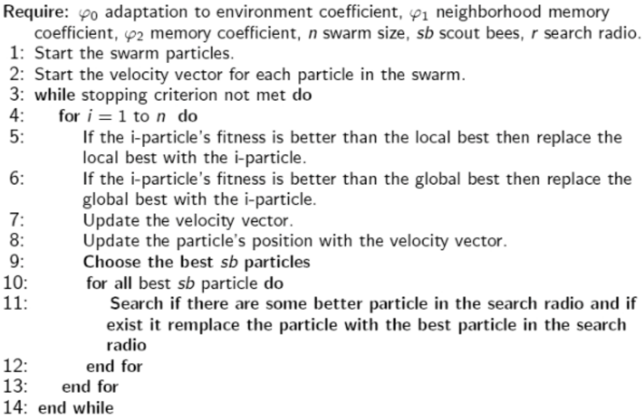
$$v_i = \varphi_0 \times v_i + \varphi_1(x_i - B_{Global}) + \varphi_2(x_i - B_{Local}) \tag{1}$$

$$s(v_i) = 1/(1 + e^{-v}) \tag{2}$$

Where:

- v_i : Velocity of the i-th particle
- φ_0: Adjustment factor to the environment
- φ_1: Memory coefficient in the neighborhood
- φ_2: Coefficient memory
- x_i : Position of the i-th particle
- B_{Global} : Best position found so far by all particles
- B_{Local} : Best position found by the i-th particle

The pseudo code for the BSO is [12]:



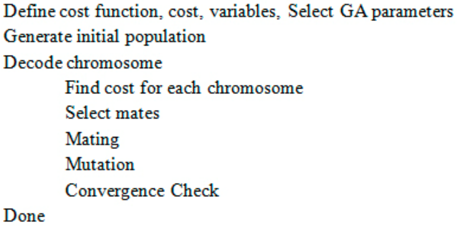
```
Require: $\varphi_0$ adaptation to environment coefficient, $\varphi_1$ neighborhood memory
    coefficient, $\varphi_2$ memory coefficient, $n$ swarm size, $sb$ scout bees, $r$ search radio.
Start the swarm particles.
Start the velocity vector for each particle in the swarm.
while stopping criterion not met do
    for $i = 1$ to $n$ do
        If the i-particle's fitness is better than the local best then replace the
        local best with the i-particle.
        If the i-particle's fitness is better than the global best then replace the
        global best with the i-particle.
        Update the velocity vector.
        Update the particle's position with the velocity vector.
        Choose the best $sb$ particles
        for all best $sb$ particle do
            Search if there are some better particle in the search radio and if
            exist it remplace the particle with the best particle in the search
            radio
        end for
    end for
end while
```

Fig. 1. The BSO Algorithm applied to a binary problem

The process of conducting particles in the process of exploration for the best from a search radius is defined as a binary operation of addition and subtraction to the particle. For example if we have the following particle 10010 with a search radius of 2, then you will be add (in binary) 1 and 2, after you subtract (using the binary representation) 1 and 2, you get the fitness with each other and if there is one that has better fitness than the current particle then is replaced because it was found a best element in the range search.

2.2 Genetic Algorithm

A genetic algorithm is basically a search and optimization technique based on the principles of genetics and natural selection. These are implemented in computer simulations where the population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem that aims to find better solutions. The operation of a simple genetic algorithm can be shown in the following pseudocode [13]:



```
Define cost function, cost, variables, Select GA parameters
Generate initial population
Decode chromosome
        Find cost for each chromosome
        Select mates
        Mating
        Mutation
        Convergence Check
Done
```

Fig. 2. Pseudocode of a binary GA

3 Proposed Solutions

The solutions of the algorithms used give us the settings (on/off) of the bulbs that should be on and off in each room, in order to provide the amount of light that the user needs to perform the activity defined. The activities taken into account in this study were: reading, computer work, relaxing, project and expose. To represent individuals or combinations of particles we will use 0's and 1's, where 0 means that the bulb is off and 1 on. Depending on which room the user is, the size of our chromosome or particle changes according to the number of outbreaks in each area selected because each room has a different number of bulbs. The second floor of the building of the Division of Postgraduate Studies and Research was taken as a test instance, which has laboratories, classrooms and cubicles, which will form part of our scenario. The parameters considered are the number of lamps in each room (lab, classroom, cubicle and corridor), lamp power, the size of the room and the activity of the user in that particular room. The Figure 3 we can see the distribution of the lights used in our test example:

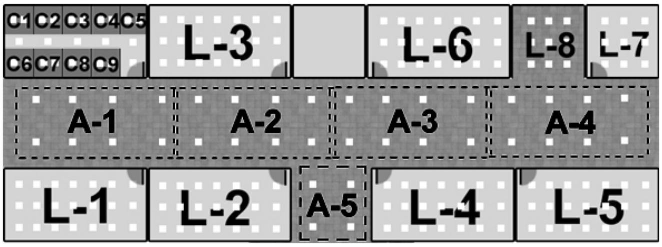
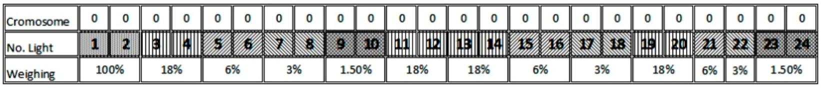


Fig. 3. Distributions of lamps on the first floor's Division of Research and Postgraduate Studies

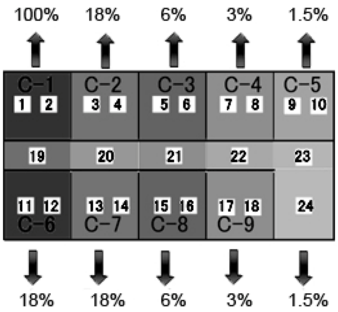
Our representation of individuals or particles depends on which room the user is, if the user is in the area of cubicles C1, C2, C3, C4, C5, C6, C7, C8 and C9, the representation of the chromosome if the user chooses C-1 in this case would be like the Figure 4:



Cromosome	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
No. Light	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Weighing	100%		18%		6%		3%		1.50%		18%		18%		6%		3%		18%		6%	3%	1.50%	

Fig. 4. Distribution and weighing of lamps in chromosome for C-1

Figure 4 shows how the representation for chromosome is if the user chooses C-1, each bit of the chromosome represents a lamp according to the number that was given to the lamps. If the user is in a cubicle, lamps of the other cubicles and corridor that are closer to cubicle selected will have a weighting based on their distance from the cube selected (this is to take into account also lamps close to the area, and which can provide some amount of light flux to the cubicle). The weighting of the lamps was based in the building with a lux meter that was placed in the middle of each of the rooms. The weights of the lamps have the following values as shown in Figure 5a:



a) b)

Fig. 5. a) Percentage of luminous flux according to weight in C-1 and enumeration of the lamps b) Enumeration of the lamps on L-1

If the user is located in any of the laboratories L1, L2, L3, L4, L5, L6, the enumeration of the lamps located in any of the laboratories listed would be as shown in the Figure 5b, because the lamps are in the same laboratory then everyone will have equal weight in terms of quantity of luminous flux. If the L-7 is currently selected then the representation of the chromosome for that laboratory would be with a size of 9 bits. If the L-8 is currently selected, takes into account that it is not a door, and therefore certain lights from the corridor can provide a certain amount of light and this condition minimize the required number of lamps lit for each activity in this area. The lamps with numeration 10, 11, 12, 13, 14 and, 15 are not located directly in L-8 but like they are closer to L-8 can proportions light flux with certain weight. The lamps 10, 11 and 12 can provide 50 % of their light flux, the lamps 13, 14 and 15 can provide 25% of their light flux, the weighting is depending on the distance of these lamps to L-8. The weights of the lamps in L-8 have shown in Figure 6.

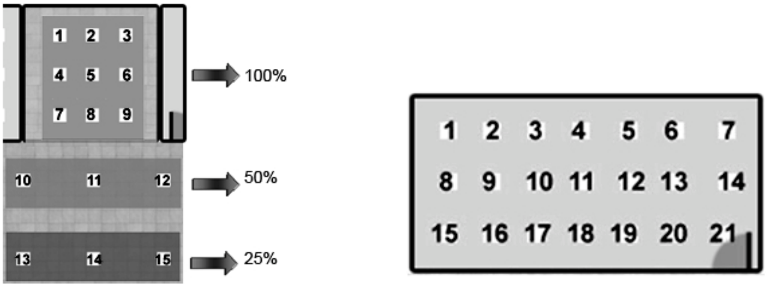


Fig. 6. Enumeration and weight of lamps on L-8

In the calculations for the interior lighting is we used the equation 3 which calculates the total luminous flux required to perform some activity taking into account the size of the area where it will work [17].

$$\Phi_T = (E \times S)/(\eta \times f_m) \tag{3}$$

Where:

- Φ_T : Total Luminous Flux (lm)
- E : Desired average luminance (lux) See Table 3 and 4
- S : Surface of the working plane (m^2)
- η : Light Output (lm/W)
- f_m : Maintenance Factor

The value of the maintenance factor is obtained from Table 1, depending on whether the environment is clean or dirty, but in this case by default takes the value of a clean environment because it is a closed and cleaned often:

Table 1. Maintenance Factor of lamps (Source: Carlos Laszlo, Lighting Design & Asoc.)

Ambient	Maintenance Factor (f_m)
Clean	0.8
Dirty	0.6

According to each lamp and depending on their power (W) Table 2 that shows the values of luminous flux and luminous efficacy. To calculate the total luminous flux is necessary also to take into account what is the desired luminance according to the place where the activity will take place. This desired luminance for each activity is shown in Table 3 and Table 4.

Table 2. Typical lamps values (Source: *Web de Tecnología Eléctrica*)

Lamp Type	Power (W)	Luminous Flux (Lm)	Luminous Efficacy (Lm/W)
Vela wax		10	
Incandescent	40	430	13,80
	100	1.300	13,80
	300	5.000	16,67
Fluorescent Lamp Compact	7	400	57,10
	9	600	66,70
Fluorescent Lamp Tubular	20	1.030	51,50
	40	2.600	65,00
	65	4.100	63,00
Lamp vapor Mercurio	250	13.500	54,00
	400	23.000	57,50
	700	42.000	60,00
Lamp Mercury Halogenous	250	18.000	72,00
	400	24.000	67,00
	100	80.000	80,00
Lamp Vapor High Pressure Sodium	250	25.000	100,00
	400	47.000	118,00
	1000	120.000	120,00

Table 3. Luminances recommended by activity and type of area (Source: *Aprendizaje Basado en Internet*)

Activities and type area	Average service luminance (lux)		
	Minimum	Recommended	Optimum
Circulation area, corridors	50	100	150
Stairs, escalators, closets, toilets, archives	100	150	200
Classrooms, laboratories	300	400	500
Library, Study Rooms	300	500	750
Office room, conference rooms	450	500	750
Works with limited visual requirements	200	300	500
Works with normal visual requirements	500	750	1000
Dormitories	100	150	200
Toilets	100	150	200
Living room	200	300	500
Kitchen	100	150	200
Study Rooms	300	500	750

Table 4. Average Service Luminance for each activity

Activity	Average service luminance (lux)
Read	750
Computer Work	500
Stay or relax	300
Project	450
Exposition	400

To calculate the fitness of each individual or particle, the sum of the luminous flux provided by each lamp on is calculated, taking into account the weighting assigned to each lamp depending on the area. The result of that sum should be as close as possible to the amount of total light output required for the activity to perform. The fitness is the subtraction of that amount and the luminous flux, and this is expressed in equation 4. For this problem we are minimizing the amount of lamps turned on according to each activity to be performed in each area, and the fittest solution is the minimum of all solutions.

$$Fitness = (\Phi_{L1} + \Phi_{L2} + \Phi_{L3} + \ \ldots \ + \ \Phi_{Ln}) - \Phi_{Total\ Required} \tag{4}$$

Where:

- Φ_{L1} : Luminous Flux of lamps (lm)
- $\Phi_{Total\ Required}$: Total Required Luminous Flux (lm)

4 Results and Comparison

The activities taken into account were: reading, computer work, relaxing, project and exposition. For this test used Laboratory L-1, L-8, C-1 and, C-3. The parameters value are based in [18].

Table 5. Input parameters using the BSO and GA

BSO		GA	
Parameter	Value	Parameter	Value
φ_0	1	Generations	100
φ_1	0.5	Population size	50
φ_2	0.8	Mutation probability	0.8
Scout bees	40%	Elitism	0.2
Iterations	100	Generations	100
Particles	50		

Table 6. Results of test instance applying BSO

Room	Activity	Mean	Standard Deviation
C-1	Read	22.20	30.76
	Computer Work	50.33	44.4
	Relax or stay	47.75	49.88
	Projection	108.35	98.72
	Exposition	34.42	27.65
C-3	Read	70.30	48.72
	Computer Work	81.53	54.53
	Relax or stay	123.80	88.5
	Projection	76.50	54.53
	Exposition	77.07	77.36
L-1	Read	553.62	1.1984E-13
	Computer Work	1019.08	919.23
	Relax or stay	2431.44	959.22
	Projection	1372.17	671.31
	Exposition	1075.26	1194.61
L-8	Leer	191.30	2.9959E-14
	Read	192.53	137.02
	Computer Work	271.52	137.03
	Relax or stay	309.78	5.9918E-14
	Projection	199.52	102.77

Table 7. Results of test instance applying GA

Room	Activity	Mean	Standard Deviation
C-1	Read	30.	26
	Computer Work	180.3	359.6
	Relax or stay	59.4	64.5
	Projection	29.05	28.9
	Exposition	34.4	35.6
C-3	Read	132.7	124.2
	Computer Work	65.9	32.8
	Relax or stay	131.6	184
	Projection	60.9	49.3
	Exposition	71.4	75.3
L-1	Read	553.6	0
	Computer Work	369	5.9918E-14
	Relax or stay	1261.4	548.1
	Projection	982.1	411.09
	Exposition	295.2	822.1
L-8	Read	191.3	0
	Computer Work	160	102.7
	Relax or stay	239	102.7
	Projection	374.7	137.03
	Exposition	232	137

Applying a Wilcoxon test to compare the results from BSO with GA, the results shown in Table 8 are found.

Table 8. Comparison of BSO with GA using the Wilcoxon Signed Rank Test

Room	Activity	BSO	GA	X-Y	Rank
C-1	Read	22.2	30.007	-7.7	6
	Computer Work	50.3	180.3	-130	15
	Relax or stay	47.7	59.4	-11.7	7
	Projection	108.3	29.05	79.2	14
	Exposition	34.4	34.4	0.0004	2
C-3	Read	70.3	132.7	-62.4	12
	Computer Work	81.5	65.9	15.6	8.5
	Relax or stay	123.8	131.6	-7.7	5
	Projection	76.5	60.9	15.6	8.5
	Exposition	77.07	71.4	5.6	4
L-1	Read	553.6	553.6	0.0031	3
	Computer Work	1019	369	649.9	17
	Relax or stay	2431.4	1261	1169.9	19
	Projection	1372.1	982.1	389.9	16
	Exposition	1075.2	295.2	779.9	18
L-8	Read	191.3	191.3	-0.0003	1
	Computer Work	192.5	160	32.4	9
	Relax or stay	271.5	239.02	32.4	10
	Projection	309.7	374.7	-65	13
	Exposition	199.5	232.02	-32.5	11

In Table 8 we have T^{+}= 70, T^{-} = 129 and according to the table of critical values of T in the Wilcoxon Signed Rank Test [16], on the N=20 with P=0.10 and a confidence level of 99.9, we have t_0=60, as for this problem we are minimizing.

If $T^{-} < t_0$ met, Then is accepted H_0 that the data have the same distribution. Then the distributions aren´t different and T^{+} is more to the right. The Genetic Algorithm is more to the left and therefore has a better performance in the minimization process.

5 Conclusions and Future Work

According to the experiments performed (based on GA and BSO) and after applying the Wilcoxon Signed Rank Test [16] the optimal results are found with the GA. The test can be shown that 12 of the activities in different rooms has optimal results for the GA, because X-Y in Wilcoxon Test has more positive number because the GA has optimal results for this minimization problem.

We can get settings for the management of bulbs in our scenario and improve our energy efficiency because the lights will turn on and off according to the different activities. In addition, the system also use the light provided by the surroundings, such as rooms and corridors.

As future research we are planning to add more input parameters, such as ventilation, and include other devices in our scenario.

References

1. Zelkha, E., Epstein, B.B.: From Devices to Ambient Intelligence: The Transformation of Consumer Electronics. In: Digital Living Room Conference (1998)
2. ISTAG Scenarios for Ambient Intelligence in Compiled by Ducatel, K., M.B. 2010 (2011)
3. Sulaiman, F., Ahmad, A.: Automated Fuzzy Logic Light Balanced Control Algorithm Implemented in Passive Optical Fiber Daylighting System (2006)
4. Boman, M., Davidsson, P., Skarmeas, N., Clark, K.: Energy saving and added customer value in intelligent buildings. In: Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (1998)
5. Akkermans, J., Ygge, F.: Homebots: Intelligent decentralized services for energy management. Ergon Verlag (1996)
6. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press (1975)
7. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks (1995)
8. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S.: The bees algorithm–a novel tool for complex optimisation problems. In: Proc 2nd Int Virtual Conf. on Intelligent Production Machines and Systems (IPROMS 2006), pp. 454–459 (2006)
9. Nieto, J.: Algoritmos basados en cúmulos de partículas para la resolución de problemas complejos (2006)
10. Sotelo-Figueroa, M.A., Baltazar, R., Carpio, M.: Application of the Bee Swarm Optimization BSO to the Knapsack Problem. In: Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Recognition Based on Biometrics. SCI, vol. 312, pp. 191–206. Springer, Heidelberg (2010), doi:10.1007/978-3-642-15111-8_12 ISBN: 978-3-642-15110-1
11. Sotelo-Figueroa, M.A., del Rosario Baltazar-Flores, M., Carpio, J.M., Zamudio, V.: A Comparation between Bee Swarm Optimization and Greedy Algorithm for the Knapsack Problem with Bee Reallocation. In: 2010 Ninth Mexican International Conference on Artificial Intelligence (MICAI), November 8-13, pp. 22–27 (2010), doi: 10.1109/MICAI.2010.32
12. Sotelo-Figueroa, M., Baltazar, R., Carpio, M.: Application of the Bee Swarm Optimization BSO to the Knapsack Problem. Journal of Automation, Mobile Robotics & Intelligent Systems (JAMRIS) 5 (2011)
13. Haupt, R.L.: Practical Genetic Algorithms (2004)
14. Hernández, J. L. (s.f.): Web de Tecnología Eléctrica. Obtenido de Web de Tecnología Eléctrica, `http://www.tuveras.com/index.html`

15. Fernandez, J.G. (s.f.): EDISON, Aprendizaje Basado en Internet. Obtenido de EDISON, Aprendizaje Basado en Internet, `http://edison.upc.edu/`
16. Woolson, R.: Wilcoxon Signed-Rank Test. Wiley Online Library (1998)
17. Laszlo, C.: Lighting Design & Asoc. (n.d.). Manual de luminotecnia para interiores. retrieved from Manual de luminotecnia para interiores, `http://www.laszlo.com.ar/manual.htm`
18. Sotelo-Figueroa, M.A.: Aplicacion de Metahueristicas en el Knapsack Problem (2010)

Modeling Prey-Predator Dynamics via Particle Swarm Optimization and Cellular Automata

Mario Martínez-Molina[1], Marco A. Moreno-Armendáriz[1],
Nareli Cruz-Cortés[1], and Juan Carlos Seck Tuoh Mora[2]

[1] Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz s/n, México D.F., 07738, México
mariomartinezmolina@live.com

[2] Centro de Investigación Avanzada en Ingeniería Industrial,
Universidad Autónoma del Estado de Hidalgo,
Carr. Pachuca-Tulancingo Km. 4.5, Pachuca Hidalgo 42184, México

Abstract. Through the years several methods have been used to model organisms movement within an ecosystem modelled with cellular automata, from simple algorithms that change cells state according to some pre-defined heuristic, to diffusion algorithms based on the one dimensional Navier - Stokes equation or lattice gases. In this work we show a novel idea since the predator dynamics evolve through Particle Swarm Optimization.

1 Introduction

Cellular Automata (CA) based models in ecology are abundant due to their capacity to describe in great detail the spatial distribution of species in an ecosystem. In [4], the spatial dynamics of a host-parasitoid system are studied. In this work, a fraction of hosts and parasites move to colonize the eight nearest neighbors of their origin cell, the different types of spatial dynamics that are observed depend on the fraction of hosts and parasitoid that disperse in each generation. Low rates of host dispersal lead to chaotic patterns. If the rate of host dispersal is too low, and parasitoid dispersal rates are very high, "crystal lattice" patterns may occur. Mid to hight rates of host dispersal lead to spiral patterns.

In [9], an individual-oriented model is used to study the importance of prey and predator mobility relative to an ecosystem's stability. Antal and Droz [1] used a two-dimensional square lattice model to study oscillations in prey and predator populations, and their relation to the size of an ecosystem. Of course, organisms have multiple reasons to move from one zone of their habitat to another, whether to scape from predation, or to search the necessary resources for survival. An example appears in [8], where predators migrate via lattice gas interactions in order to complete their development to adulthood.

In this work we show a CA model of a theoretical population, where predator dynamics evolve through Particle Swarm Optimization (PSO). Each

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 189–200, 2011.

season, predators search the best position in the lattice according to their own experience and the collective knowledge of the swarm, using a fitness function that assigns a quality level according to local prey density in each site of the lattice. To the best of our knowledge, such approach has never been used to model predator dynamics in an spacial model. The results show oscillations typical of Lotka -Volterra systems, where for each increase in the size of the population of predators, there is a decrease in the size of the population of preys.

2 Background

2.1 Cellular Automata

CA are dynamical systems, discrete in time and space. They are adequate to model systems that can be described in terms of a massive collection of objects, known as cells, which interact locally and synchronously. The cells are located on the d-dimensional euclidean lattice $L \subseteq \mathbb{Z}^d$. The set of allowed states for each cell is denoted by Q. Each cell changes its state synchronously at discrete time steps according to a local transition function $f : Q^m \to Q$, where m is the size of the d-dimensional neighborhood vector N defined as:

$$N = (\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3, \ldots, \boldsymbol{n}_m) \tag{1}$$

where $\boldsymbol{n}_i \in \mathbb{Z}^d$. Each $\boldsymbol{n}_i$ specifies the relative locations of the neighbors of each cell [6], in particular, cell $\boldsymbol{n}$ has coordinates $(0, 0, \ldots, 0)$ and neighbors $\boldsymbol{n} + \boldsymbol{n}_i$ for $i = 1, 2, \ldots, m$. A configuration of a d- dimensional cellular automaton is a function:

$$c : \mathbb{Z}^d \to Q$$

that assigns a state to each cell. The state of cell $\boldsymbol{n} \in \mathbb{Z}^d$ at time t is given by $c^t(\boldsymbol{n})$, the set of all configurations is $Q^{\mathbb{Z}^d}$. The local transition function provokes a global change in the configuration of the automata. Configuration c is changed into configuration c', where for all $\boldsymbol{n} \in \mathbb{Z}^d$:

$$c'(\boldsymbol{n}) = f\left[c(\boldsymbol{n} + \boldsymbol{n}_1), c(\boldsymbol{n} + \boldsymbol{n}_2), \ldots, c(\boldsymbol{n} + \boldsymbol{n}_m)\right] \tag{2}$$

The transformation $c \mapsto c'$ is the global transition function of the cellular automaton, defined as:

$$G : Q^{\mathbb{Z}^d} \to Q^{\mathbb{Z}^d} \tag{3}$$

In a two dimensional cellular automaton the Moore neighborhood is often used, such neighborhood can be generalized as the d-dimensional M_r^d neighborhood [6] defined as:

$$(n_{i_1}, n_{i_2}, \ldots, n_{i_d}) \in \mathbb{Z}^d \text{ where } |n_{i_j}| \leq r \text{ for all } j = 1, 2, \ldots, d \tag{4}$$

2.2 Particle Swarm Optimization

Particle Swarm Optimization is a bio-inspired algorithm based on the collective behavior of several groups of animals (flocks, fish schools, insect swarms, etc) [5]. The objective of PSO is the efficient exploration of a solution space, each individual in a 'community' is conceptualized as a particle moving in the hyperspace. Such particles have the capacity to 'remember' the best position they have been in the solution space, furthermore in the global version of PSO, the best position found thus far is known to every particle of the swarm.

The position X_i^t of every particle in the swarm is updated in discrete time steps according to the next equations:

$$V_i^{t+1} = \omega V_i^t + k_1 r_1 \left(P_i^t - X_i^t\right) + k_2 r_2 \left(P_g^t - X_i^t\right) \tag{5}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{6}$$

where V_i^t is the velocity vector at time t associated to particle i, the constants k_1 and k_2 determine the balance between the experience of each individual (the cognitive component) and the collective knowledge of the swarm (the social component) respectively [2]. $r_1 \in [0,1]$ and $r_2 \in [0,1]$ are random variables with a uniform distribution. The best position found by the particle i is denoted by P_i, similarly the best position found by the swarm is denoted by P_g. The term ω is known as inertia weight and serves as a control mechanism to favor exploration of the solution space or exploitation of known good solutions. In [7] it is suggested to start the algorithm with $\omega = 0.9$ and linearly decrement it to $\omega = 0.4$, thus at the beginning of the algorithm exploration is favoured, and at the end exploitation is enhanced. Figure 1 shows the position updating scheme according to equations 5 and 6.

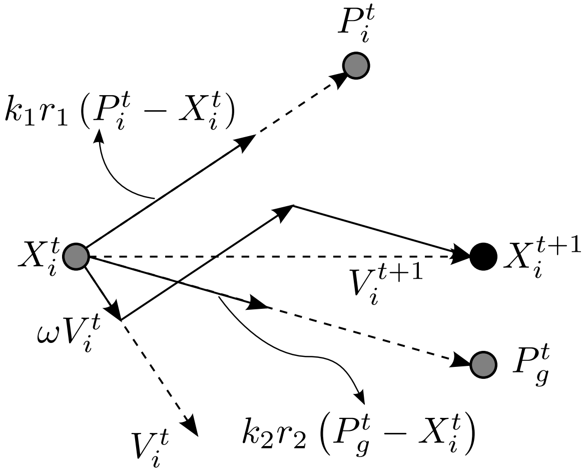

P_i^t
$k_1 r_1 \left(P_i^t - X_i^t\right)$
X_i^t
X_i^{t+1}
V_i^{t+1}
ωV_i^t
P_g^t
V_i^t
$k_2 r_2 \left(P_g^t - X_i^t\right)$

Fig. 1. Position updating scheme in PSO [11]

3 Proposed Model

Our model describes a theoretical ecosystem, where a sessile prey and a predator live. The individuals of the prey species compete locally with other members of their own species (interspecific competence), prey reproduction is a local process. In order to secure their own future, and that of their progeny, predators migrate each season from zones low on resources (preys) to zones highly abundant in food, just as in the case of preys, predators reproduce locally.

The space in which species live and interact is represented by the lattice $L \subset \mathbb{Z}^2$, periodic boundaries have been implemented, i.e. the cellular space takes the form of a torus. The set of allowed states for each cell is:

$$Q = \{0, 1, 2, 3\} \tag{7}$$

where:

- 0 is an empty cell.
- 1 is a cell inhabited by a prey.
- 2 is a cell inhabited by a predator.
- 3 is a cell containing a prey and a predator at the same time.

Both preys and predators, obey a life cycle that describes their dynamics in a generation. Predator dynamics are modelled through the next rules:

1. **Migration.** During this stage, predators move within the cellular space according to their own experience and the collective knowledge of the swarm.
2. **Reproduction.** Once the migration is complete, each predator produces new individuals at random inside a Moore neighborhood of radius two.
3. **Death.** Predators in cells lacking a prey die by starvation.
4. **Predation.** Preys sharing a cell with a predator die due to predator action.

On the other hand, the life cycle of preys is modelled under the following assumptions:

1. Intraspecific competence. Preys die with a probability proportional to the number of individuals of the prey species surrounding them, this rule uses a Moore neighborhood of radius 1. If $c^t(\boldsymbol{n}) = 1$, then the probability of death ($c^{t+1}(\boldsymbol{n}) = 0$) is given by:

$$\rho\,(death) = \frac{\alpha x}{m} \tag{8}$$

   where:
   - $\alpha \in [0, 1]$ is the intraspecific competence factor, which determines the intensity of competence exercised by preys in the neighborhood of cell $\boldsymbol{n}$.
   - x is the number of preys in the neighborhood of cell $\boldsymbol{n}$.
   - $m = |N|$.
2. Reproduction. Like predators, preys spawn new individuals at random in a Moore neighborhood of radius 2.

Each stage in preys and predators dynamics occurs sequentially. They form a cycle that defines one generation in their life, such cycle is:

1. Intraspecific Competence of preys.
2. Migration of predators.
3. Predators reproduction.
4. Predators death.
5. Predation
6. Prey reproduction.

As this cycle suggests, at each stage the rule applied to cells changes accordingly.

4 PSO as a Migration Algorithm

The main contribution in this work is to utilize a PSO algorithm as a mechanism to model the migration of predators, that is, predators change their position according to PSO. Some important differences in the use of PSO as a migration algorithm and its use in numerical optimization are:

- **Fitness.** In numerical optimization, it is common to use the same function to optimize as a mean to obtain a measure of a solution's fitness. In the proposed model, the solution space is the lattice of the CA, so each cell represents a candidate solution to the problem of finding the necessary resources for survival and procreation. Since the landscape of an ecosystem changes continuously, it is impossible to speak of an absolute best cell, instead each predator moves to the known "good" enough zones and exploits them. Once depleted, predators migrate to search for new zones for feeding and procreation, so instead of aiming for a global optima, predators exploit known local optima.
- **Solution space.** As stated before, the lattice takes the form of a torus and represents the solution space in which each particle of the swarm moves. Thus the movement of a particle can take a predator from one edge of the lattice to the other, this favours exploration.
- **Swarm size.** In our model each particle is also a predator, in consequence, they can die, and they can reproduce, this changes the size of the swarm in each generation.

Since the model is discrete in space, the update of a particle's position simply determines the location to which the particle moves to. Consequently the cell from which the predator initiates its migration could go through the following state changes:

$$c^t(\boldsymbol{n}) = 2 \rightarrow c^{t+1}(\boldsymbol{n}) = 0$$
$$c^t(\boldsymbol{n}) = 3 \rightarrow c^{t+1}(\boldsymbol{n}) = 1$$

Similarly, the cell in which the predator ends its migration could experience the next state transitions:

$$c^t(\boldsymbol{n}) = 0 \rightarrow c^{t+1}(\boldsymbol{n}) = 2$$
$$c^t(\boldsymbol{n}) = 1 \rightarrow c^{t+1}(\boldsymbol{n}) = 3$$

As a measure of a particle's fitness, we use prey density in the neighborhood N of each cell, thus, a cell with more preys in its neighborhood is a better location than a cell with less preys in its neighborhood.

4.1 Migration Process

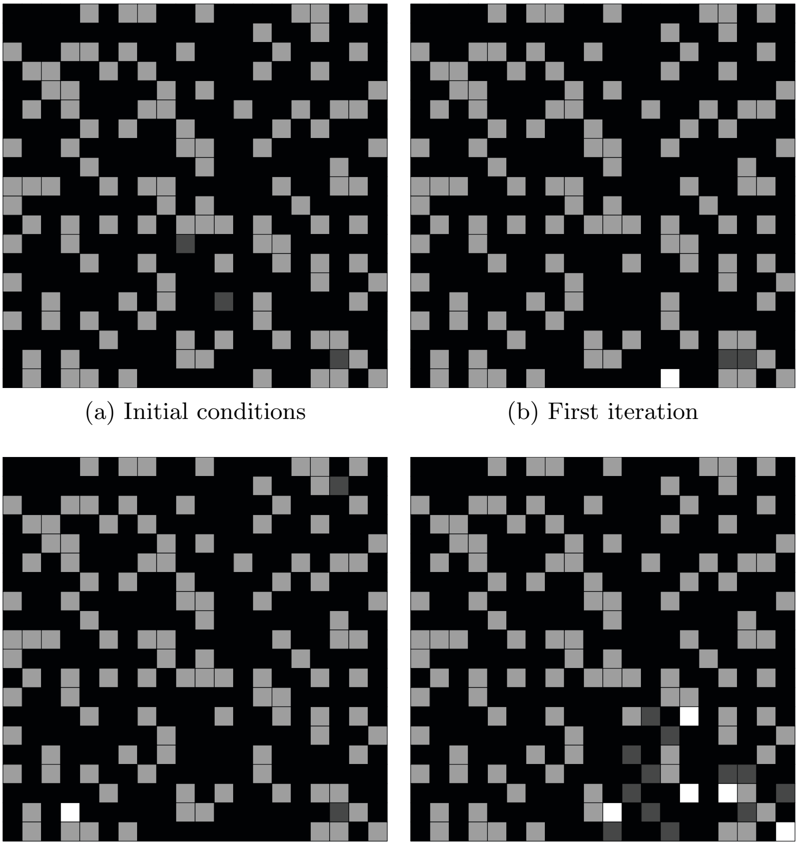
As stated in section 3, migration takes place after the competence of preys. At the beginning of each migration, particles determine the fitness of their current position (by measuring prey density in its neighborhood), and set their best known position. Using this information, the best known position of the swarm is set. After this initialization step, migration proceeds as follows:

1. The velocity vector of each particle is updated according to equation 5, the magnitude of which depends on the values taken by parameters ω, k_1, k_2, r_1 and r_2.
2. Each particle moves to its new position by adding the vector V_i^{t+1} to its current position X_i^t.
3. The new neighborhood is explored and if necessary, both the best known position of each particle P_i^t, and the best position of the swarm (P_g^t) are updated.
4. The value of the inertia weight ω is adjusted.

This process is repeated 5 times, to ensure a good search in the proximity of the zones known by the swarm and by each individual particle. Figure 2 shows the migration of a swarm of 3 particles through PSO. The states of each cell are shown with the next color code:

- **Black:** empty cell - state 0
- **Light gray:** prey - state 1
- **Dark gray:** predator - state 2
- **White:** cell inhabited by a prey and a predator at the same time.

Figure 2a shows initial conditions, of the 3 individuals, the one located at the bottom - right is the one with the best fitness, so the other two will move in that direction (Figures 2b and 2c). When predators end their migration, they reproduce, so by migrating to zones with a high prey density, not only they have a better chance of survival, but their offspring too.



(a) Initial conditions

(b) First iteration

(c) Second iteration

(d) Reproduction

Fig. 2. Migration through PSO

5 Comparison with Lotka - Volterra Systems

The growth of a population in the absence of predators and without the effects of intraspecific competence can be modeled through the differential equation [3]:

$$\frac{dZ}{dt} = \gamma Z \tag{9}$$

where:

- Z is the size of the population.
- γ is the population's rate of growth.

However, when predation is taken into account, the size of the population is affected proportionally to the number of predator-prey encounters, which depend on the size of the populations of preys (Z) and predators (Y). Since predators are not perfect consumers, the actual number of dead preys depends on the efficiency of the predator. Let a be the rate at which predators attack preys, thus the rate of consumption is proportional to aZY, and the growth of the population is given by:

$$\frac{dZ}{dt} = \gamma Z - aZY \tag{10}$$

Equation 10 is known as the Lotka-Volterra prey equation. In the absence of preys, the population of predators decay exponentially according to:

$$\frac{dY}{dt} = -sY \tag{11}$$

where s is the predator mortality rate. This is counteracted by predator birth, the rate of which depend on only two things: the rate at which food is consumed, aZY, and the predator's efficiency h, predator birth rate is $haZY$, thus:

$$\frac{dY}{dt} = haZY - sY \tag{12}$$

Equation 12 is known as the Lotka-Volterra predator equation. Figure 3 shows the dynamics of an ecosystem ruled by equations 10 and 12.

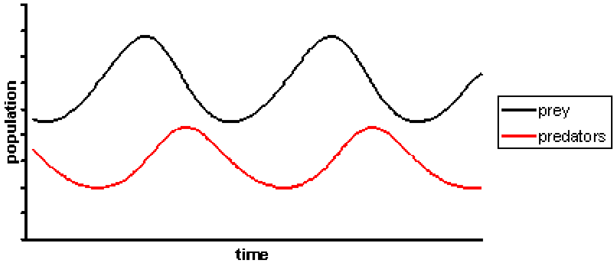


Fig. 3. Lotka - Volterra prey - predator dynamics

The Lotka-Volterra equations show periodic oscillations in predator and prey populations. This is understandable given the next reasoning: when there is an abundant number of preys, the food consumption by predators increases, and thus the number of predators grows. Due to this fact, the number of prey diminishes, and so does the food available to predators, which increase predator mortality. The death of predators allows a new increase in the population of preys, and the process begins anew. An excellent review of lattice based models that give new perspectives on the study of oscillatory behavior in natural populations appears in [10].

It is possible to simulate the behavior of Lotka-Volterra equations through the proposed model, most of the parameters of these equations are indirectly taken into account in such model, e. g., predator efficiency depends on whether predators have a successful migration or not. To simulate the behavior of equations 10 and 12, the next parameters are used.

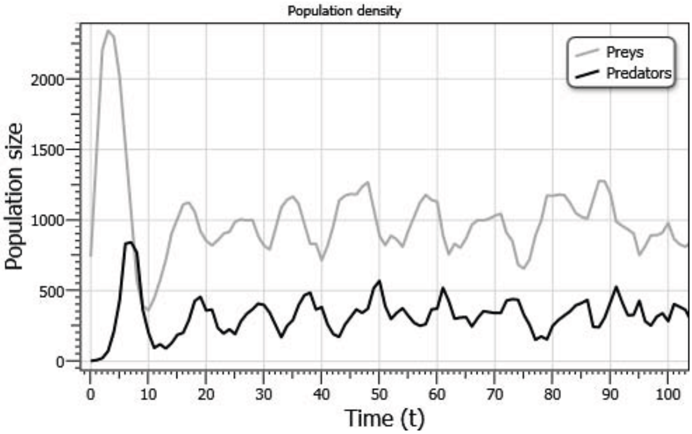


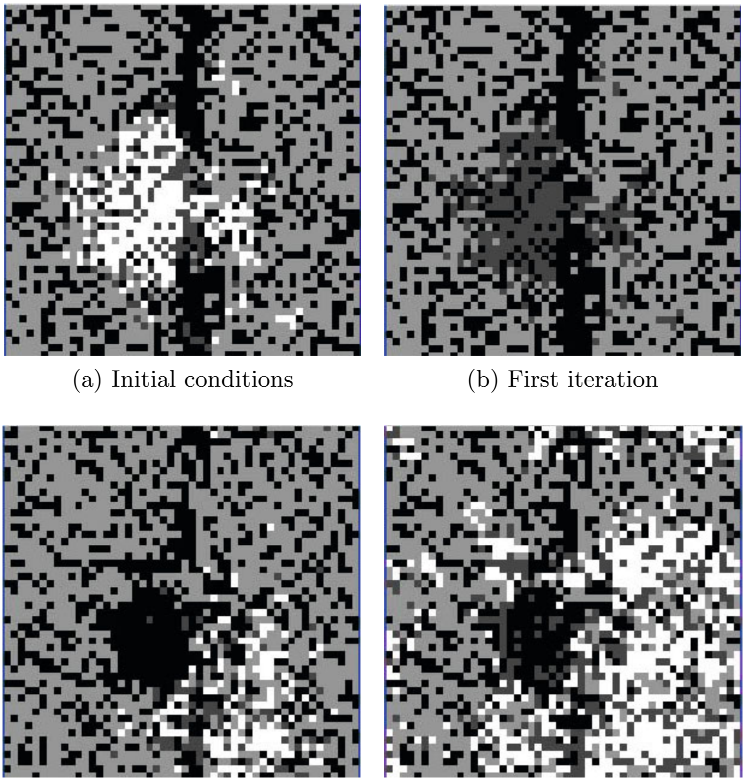
Fig. 4. Prey - predator dynamics through PSO in a CA

- Size of the lattice: $50 \times 50 = 2500$ cells.
- Initial prey percentage: 30%
- Intraspecific competence factor: $\alpha = 0.3$. If this parameter is too high, most of the ecosystem will be composed of small "patches" of preys separated by void zones, in consequence only a fraction of predators will survive the migration.
- Mean offspring of each prey: 3 individuals.
- Swarm's initial size: 3 particles.
- Mean offspring of each predator: 5 individuals. A high predator reproductive rate would lead to over-exploitation of resources, in consequence there is a chance that predators will go extinct.
- $k_1 = 2.0$ and $k_2 = 2.0$.
- Initial inertia weight $\omega = 0.9$ and Final inertia weight $\omega = 0.4$
- $|V_{max}| = \frac{\text{lattice width}}{3}$

Figure 4 shows the dynamics of the proposed model, oscillations obeying the abundance cycles of prey and predators are shown. Figure 5a shows a swarm about to begin a migration, after feeding on preys (Figure 5b), there is a wide empty zone where most of the cells have a fitness equal to zero. In order to survive, predators move to "better" zones. In Figure 5c most of the swarm has moved away from the empty zone (differences in the distribution of prey are due to the process of competence and reproduction of the past iteration) to zones with a higher density of preys. The migration of predators allows the colonization of the previously predated zone, meanwhile recently attacked zones will be reflected in a decrease in the population of preys (Figure 5d).

5.1 Extinction

A small population of predators with a high reproductive capacity might lead to over-exploitation of resources (Figure 6a). Figure 6d shows the results of a



(a) Initial conditions

(b) First iteration

(c) Second iteration

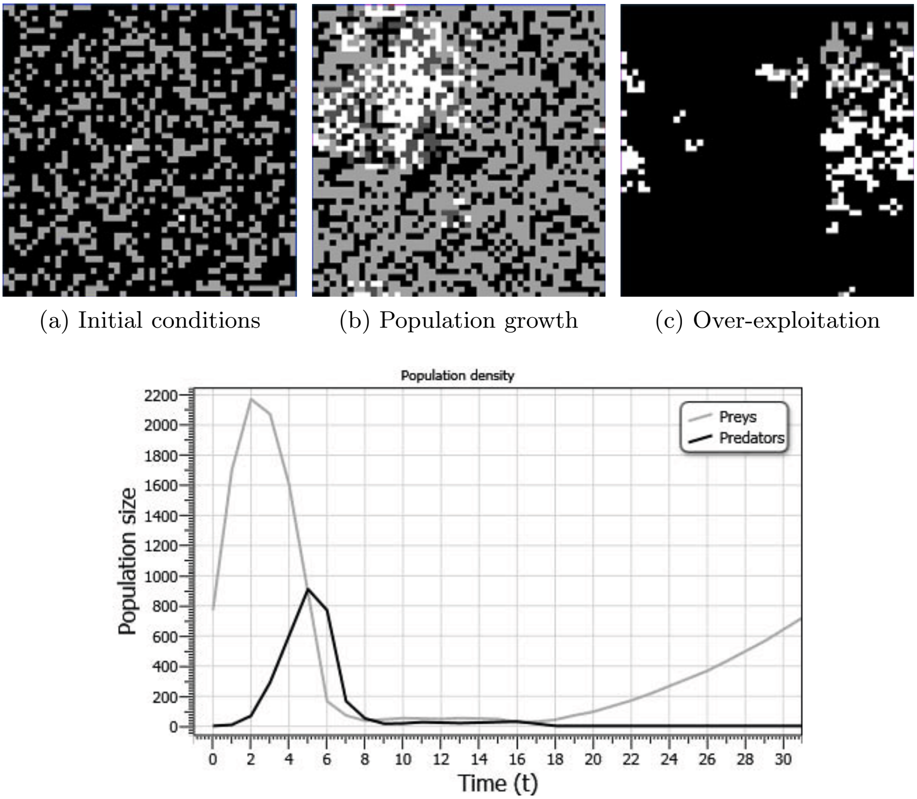
(d) Reproduction

Fig. 5. Spatial dynamics in the proposed model

simulation where each predator has a mean offspring of 15 individuals. As the size of the swarm grows (Figure 6b), bigger patches of preys are destroyed, and eventually migration becomes too difficult for most of the predators (Figure 6c). Each passing generation, the number of surviving predators decreases, until the whole population becomes extinct.

5.2 Discussion

There are other experiments that are worth discussing. It is possible to adjust the range of local search by altering the value of the inertia weight ω. By setting "high" initial and final values for this parameter, it is possible to increase the radius of local search, particles explore a wider area in the vicinity of known good zones. In consequence, most particles become disperse, and if resources are abundant, a higher predation efficiency is achieved; but if resources are sparse, the search will lead them to zones devoid of resources, and most of them will die. On the other hand, "smaller" values for the inertia weight will produce a very compact swarm specialized in local exploitation of resources.



(a) Initial conditions

(b) Population growth

(c) Over-exploitation

(d) Extinction dynamics

Fig. 6. Predators extinction

It is necessary to determine the relation between the size of the lattice, and the long term dynamics of the model. Other works [12] [1], have reported oscillations of the Lotka-Volterra type only when the size of an ecosystem is “large enough”.

6 Conclusions and Future Work

We have presented a CA based model of a theoretical ecosystem where predators migrate through PSO in order to find resources. Here we have presented the simplest implementation of PSO, yet the results are promising, it is certainly possible to establish some other fitness measures, thus it would be possible for organisms to move according to some other factors, i.e. temperature, pollution, chemical factors, etc. Of course, it is necessary to analyse the full dynamics of the model, in order to establish its strengths and weaknesses. A substantial improvement of the model would be the implementation of the local PSO, this will allow individuals to react to the information received from local members of the swarm in a finite neighborhood, thus allowing a more realistic modeling, where individuals only have access to the information of their nearest neighbors.

Acknowledgements. We thank the support of Mexican Government (SNI, SIP-IPN, COFAA-IPN, PIFI-IPN and CONACYT). Nareli Cruz-Cortés thanks CONACYT through projects 132073 and 107688 and SIP-IPN 20110316.

References

1. Antal, T., Droz, M.: Phase transitions and oscillations in a lattice prey-predator model. Physical Review E 63 (2001)
2. Banks, A., Vincent, J., Anyakoha, C.: A review of particle swarm optimization Part I: background and development. Natural Computing 6(4) (2007)
3. Begon, M., Townsend, C.R., Harper, J.L.: Ecology: From Individuals to Ecosystems, 4th edn. Blackwell Publishing (2006)
4. Comins, H.N., Hassell, M.P., May, R.M.: The spatial dynamics of host-parasitoid systems. The Journal of Animal Ecology 61(3), 735–748 (1992)
5. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
6. Kari, J.: Theory of cellular automata: a survey. Theoretical Computer Science 334, 3–33 (2005)
7. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence, 1st edn. Morgan Kauffman (2001)
8. van der Laan, J.D., Lhotka, L., Hogeweg, P.: Sequential predation: A multi-model study. Journal of Theoretical Biology 174, 149–167 (1995)
9. Mccauley, E., Wilson, W.G., de Roos, A.M.: Dynamics of age-structured and spatially structured predator-prey interactions: Individual-based models and population-level formulations. American Naturalist 142(3), 412–442 (1993)
10. Pekalski, A.: A short guide to predator-prey lattice models. Computing in Science and Engineering 6(1) (2004)
11. Shi, Y., Liu, H., Gao, L., Zhang, G.: Cellular particle swarm optimization. In: Information Sciences - ISCI (2010)
12. Wolff, W.F.: Microinteractive predator-prey simulations. Ecodynamics: Contributions to Theoretical Ecology pp. 285–308 (1988)

Topic Mining Based on Graph Local Clustering

Sara Elena Garza Villarreal[1] and Ramón F. Brena[2]

[1] Universidad Autónoma de Nuevo León, San Nicolás de los Garza NL 66450, Mexico
sara.garzavl@uanl.edu.mx
[2] Tec de Monterrey, Monterrey NL 64849, Mexico
ramon.brena@itesm.mx

Abstract. This paper introduces an approach for discovering thematically related document groups (a topic mining task) in massive document collections with the aid of graph local clustering. This can be achieved by viewing a document collection as a directed graph where vertices represent documents and arcs represent connections among these (e.g. hyperlinks). Because a document is likely to have more connections to documents of the same theme, we have assumed that topics have the structure of a *graph cluster*, i.e. a group of vertices with more arcs to the inside of the group and fewer arcs to the outside of it. So, topics could be discovered by clustering the document graph; we use a *local* approach to cope with scalability. We also extract properties (keywords and most representative documents) from clusters to provide a summary of the topic. This approach was tested over the Wikipedia collection and we observed that the resulting clusters in fact correspond to topics, which shows that topic mining can be treated as a graph clustering problem.

Keywords: topic mining, graph clustering, Wikipedia.

1 Introduction

In a time where sites and repositories become flooded with countless information (which results from the interaction with constantly evolving communication platforms for the usual), data mining techniques undoubtedly give us a hand, and they do this by extracting valuable knowledge that is not visible at a first glance. A challenging—yet interesting—sub-discipline of this domain concerns *topic mining*, i.e. the automatic discovery of themes that are present in a document collection. Because this task mainly serves the purpose of information organization, it has the potential for leveraging valuable applications, such as visualization and semantic information retrieval.

While topic mining is usually related to content (text), Web collections—which we will take as our case study for the present research —offer a tempting alternative: *structure* (hyperlinks). This information source not only is language-independent and immune to problems like polysemy[1] or assorted writing styles,

[1] Language presents two types of ambiguity: *synonymy* and *polysemy*. The former refers to different words having the same meaning, and the latter refers to a word having different meanings.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 201–212, 2011.

but has also led to the development of successful algorithms such as Google's PageRank. In that sense, there is more to structure than meets the eye.

Our primary hypothesis is that *topic mining* (where by "topic" we mean a thematically related document group) is *realizable* in a document collection by using *structure*. To achieve the former, it is necessary to first view the collection as a directed graph where vertices are given by documents and arcs are given by hyperlinks. If we consider that a topical document group will have more hyperlinks to the inside of the group and fewer hyperlinks to the outside, then a topic resembles a *graph cluster*; it is thus possible to treat topic mining as a graph clustering problem.

Being aware that Web collections tend to be large (specially since the inception of social Web 2.0 technologies), our clustering method is inspired on *graph local clustering* (which we will refer to as "GLC" for short); this technique explores the graph by regions or neighborhoods to cope with considerable sizes. Also, even though document clustering can be considered as the central part of our topic mining approach, we consider as well the extraction of topic *properties*, i.e. semantic descriptors that help to summarize a topic.

Our main contributions consist of:

1. A topic mining approach based on graph local clustering and the extraction of semantic descriptors (properties).
2. A set of topics extracted from Wikipedia (a massive, popular Web collection).
3. Evidence of the effectiveness of the approach

The remainder of this document is organized as follows: Section 2 presents relevant background, Section 3 describes our topic mining approach, Section 4 discusses experiments and results, Section 5 introduces related work, and finally Section 6 presents conclusions and future work.

2 Background

The current section discusses necessary mathematical notions and foundational literature.

2.1 Mathematical Notions

In an unweighted graph $G = (V, E)$, a graph cluster[2] consists of a vertex group $\mathcal{C}$ whose members (either individually or collectively) share more edges among themselves and fewer edges with other vertices in the graph. More formally, the *internal degree* of $\mathcal{C}$ is greater than its *external degree*; the internal degree is given by the amount of edges that have both endpoints in $\mathcal{C}$:

$$\deg_{\text{int}}(\mathcal{C}) = |\,\{\langle u, v\rangle : \langle u, v\rangle \in E, u \wedge v \in \mathcal{C}\}\,|. \tag{1}$$

[2] In complex network literature a graph cluster is known as *community*, and graph clustering is referred to as *community detection*. On the other hand, "clustering" does not imply grouping in complex network vocabulary, but rather transitivity among groups of vertices.

Conversely, the external degree is given by the amount of edges that have only one endpoint in $\mathcal{C}$:

$$\deg_{\text{ext}}(\mathcal{C}) = |\{\langle u, v\rangle : \langle u, v\rangle \in E, u \in \mathcal{C} \wedge v \notin \mathcal{C}\}|. \tag{2}$$

An alternate way of formally describing graph clusters implies the use of *relative density* (denoted by ρ), i.e. the internal edge ratio (note that $\deg(\mathcal{C}) = \deg_{\text{int}}(\mathcal{C}) + \deg_{\text{ext}}(\mathcal{C})$):

$$\rho(\mathcal{C}) = \frac{deg_{\text{int}}(\mathcal{C})}{\deg(\mathcal{C})}. \tag{3}$$

It is important to highlight that if the graph is directed (which is actually our case), by definition the out-degree $\deg_{\text{out}}(\mathcal{C})$ is used as denominator instead of $\deg(\mathcal{C})$ [21].

By utilizing relative density, we can define a graph cluster as a group $\mathcal{C}$ where $\rho(\mathcal{C}) \geq 0.5$.

2.2 Foundational Literature

Our approach is a point that lies in three dimensions: (i) topic mining, (ii) Web structure mining, and (iii) Wikipedia mining. For the sake of conciseness, a succinct description accompanied by seminal work shall be provided for each dimension. For a deeper review of these, please refer to the doctoral thesis by Garza [5].

With regard to topic mining, it embraces a wide variety of methods that can be classified according to (a) topic *representation* (label, cluster, probabilistic model, or a mixture of these) or (b) mining *paradigm* (modeling [8], labeling [20], enumeration [4], distillation [10], combinations [12]).

Web structure mining discovers patterns given hyperlink information; approaches for group detection specifically can be classified with respect to three central tasks: (a) resource discovery [10], (b) data clustering [7], and (c) graph clustering [4].

Finally, Wikipedia mining focused on semantics extraction comprises approaches that may be hard (manual) or soft (automatic), and approaches that either use Wikipedia as source or as both destination and source. An important contribution on this context is DBPedia [1].
For proper related work, please refer to Section 5.

3 Topic Mining Approach

Our topic mining approach views topics as objects consisting of a *body* and a *header*; while the former contains a set or *cluster* of member documents, the latter concentrates summary features— extracted from the body —that we will refer to simply as *properties*; the two properties we focus on are a topic *tag* (set of keywords) and a set of *representative documents* (subset of the document cluster). A topic $\mathcal{T}_i$ can thus be formally viewed as

$$\mathcal{T}_i = (\mathcal{C}_i, P_i), \tag{4}$$

where $\mathcal{C}_i$ stands for the document cluster and P_i for the properties. To illustrate this formal notion, let us present a short (created) example for the "Lord of the Rings" trilogy:

$$\begin{aligned}\mathcal{T}_{\text{lotr}} = (&\\ &\{\text{peterjackson, lotr1, lotr2, lotr3, frodo, gandalf}\},\\ &(\\ &\ \{\text{“lord”, “rings”, “fellowship”, “towers”, “king”}\},\\ &\ \{\text{lotr1, lotr2, lotr3}\}\\ &)\\ ).&\end{aligned}$$

With regard to the operations used for discovering and describing a topic, we employ *graph clustering* to extract a topic's body and then apply *ranking and selection* methods over this body to extract the header. Because a clustering method produces all clusters for a given collection at once, every topic body would actually be extracted first; after this is done, properties for the bodies are calculated. In that sense, let us start by explaining the utilized clustering method, which is also the central part of our topic mining approach.

3.1 Graph Local Clustering for Topic Mining

The first and central part of our topic mining approach involves document clustering; our specific clustering method is inspired in *graph local clustering* ("GLC")— a strategy that detects groups (where each group starts from a seed vertex) by maximizing a cohesion function [21,18,13,11,3]. Let us first describe the motivation that led to the selection of this strategy and, afterwards, explain the method itself.

Our basic assumption is that *topics have the structure of a graph cluster*; in that sense, we are considering as a topic any document group with more connections to the inside of the group than to the outside of it. To show that such an assumption is in fact intuitive, let us consider, for example, an online article about basketball: it seems more natural to think of this article as having more hyperlinks towards and from articles like "Michael Jordan" and "NBA" than to or from "mathematics" or "chaos theory". In other words, it seems logical for a document to have a greater amount of links (connections) to other documents on the same theme than to documents on different ones. As additional motivation, a higher content similarity within short link distances— a similar notion to ours —has been empirically proven on the Web [14]. So, on one hand, we require a to develop a graph clustering method.

On the other hand, *scalability* and the need for a mechanism that detects *overlapping* groups imposes constraints on our graph clustering method. With respect to the first issue, when working with massive document collections on the scale of hundreds of thousands and links that surpass the quantity of millions (typical on the Web), size does matter. In that sense, we have to rely on a strategy with the inherent capability for handling large graphs. Moreover, topics are overlapping structures by nature, since documents may belong to more than one topic at a time. Taking all of this into account, we can follow a *local* strategy, which takes not the whole graph at once but rather works on smaller sub-graphs; furthermore, the GLC strategy (as we will see later) allows the independent discovery of individual clusters, thus allowing the detection of overlapping groups.

Clustering Algorithm. Our GLC-based algorithm corresponds to a constructive, bottom-up approach that repeatedly tries to find a graph cluster out of a starting vertex or group of vertices (called "seed") by iteratively adding vertices in the vicinity of the current cluster (which initially contains only the seed). The addition of a new vertex at each step improves a current cohesion value in the fashion of *hill-climbing* (relative density being the natural choice for the function to obtain the cohesion value).

The following (general) skeleton represents the clustering method:

1. Choose a starting vertex (seed) that has not been explored.
2. *Given this initial vertex, find a cluster of vertices that produces a cohesion peak.*
3. Discard for exploration those vertices that are part of the cluster previously found.
4. Repeat until there are no vertices left to explore.

From this skeleton, we can see that step 2 by itself constitutes the discovery of *one* cluster, while the rest of the steps (1,3,4) describe the scheduling process used to select seeds. Formally, we could represent the method in terms of two functions: a construction function

$$F(S_i, \phi) = \mathcal{C}_i \tag{5}$$

where S_i represents the input seed, ϕ is a set of tunable parameters, and $\mathcal{C}_i$ is the resulting cluster (see also Algorithm 1) and a scheduling function

$$\begin{aligned}\chi(\mathbb{S}, \psi) &= \mathbb{C} \\ &= F_{s_i \in \mathbb{S}}(S_i, \phi), \forall S_i\end{aligned} \tag{6}$$

where $\mathbb{S}$ is a list of seed sets, ψ concerns a parameter that indicates seed ordering and selection, and $\mathbb{C}$ is the produced clustering. Other components of the clustering algorithm include a vertex *removal procedure* (carried out after all additions to the cluster have been done).

Algorithm 1. GLC-based algorithm.

Description: Receives as input a seed S (initial set of documents) and returns a cluster $\mathcal{C}_i$. A new element is added to the cluster at each iteration by choosing the candidate n_j that yields the first relative density improvement; each time an element becomes part of the cluster, its neighbors become candidates for inclusion at the next iteration. When relative density can no longer be increased or a specified time limit is up, the algorithm stops. Finally, element removal is carried out as post-processing.

```
function DISCOVER-GLC-TOPIC($S$)
    $\mathcal{C}_i \leftarrow S$
    $N \leftarrow$ CREATE-NEIGHBORHOOD($\mathcal{C}_i$)
    repeat
        $\rho_{\text{curr}} \leftarrow \rho(\mathcal{C}_i)$

        while $\neg$ foundCandidate $\wedge$ (more neighbors left to explore) do
            $n_j \leftarrow$ next neighbor from $N$

            if $\rho(\mathcal{C}_i \cup n_j) > \rho_{\text{curr}}$ then
                add $n_j$ to $\mathcal{C}_i$
                UPDATE-NEIGHBORHOOD($N, n_j$)
                 foundCandidate = true
            end if

        end while

        $\rho_{\text{new}} \leftarrow \rho(\mathcal{C}_i)$
    until ($\rho_{\text{new}} = \rho_{\text{curr}}$) $\vee$ time limit is reached

    $\mathcal{C}_i \leftarrow$ REMOVAL($\mathcal{C}_i$)
    return ($\mathcal{C}_i$)
end function
```

Let us note that, at clustering time, the final ρ value for a cluster is irrelevant; nevertheless, all clusters with $\rho < 0.5$ are eliminated from the final clustering. Because a vertex is never prevented from appearing in more than one cluster (i.e. its construction is independent from others, and this enables overlapping cluster discovery), we assume that, even when it could appear in a weak (low density) cluster, it might also get into a surviving group (graph cluster). We also consider that, when clusters have $\rho < 0.5$, there is no sufficient evidence to presume that they are topics.

The presented algorithm has a worst case complexity of $O(n^3)$, as it consists of three nested cycles (search over the neighborhood for element addition is done every time a cluster attempts to grow, and this procedure is repeated for every seed of the seed list). However, this worst case can be considered as rare, mainly because the approach works in such a way that an increase on the number of repetitions for one cycle implies a decrease in the number of repetitions for another one. In that sense, the worst scenario would be given by an *unclusterable* graph, e.g. a complete unweighted graph. For a deeper explanation, please refer to Garza's thesis.

3.2 Properties

The second part of the topic mining approach relates to property extraction. As previously mentioned, we focus on two topic properties: a descriptive *tag* (composed by a set of keywords) and a subset of *representative documents*, the former being used to name the topic and the latter being used to capture its essence. The methods we use for calculating one and the other have a common backbone: rank according to some relevance metric and select the top k elements (words or documents, depending on the case).

For topic tag generation, the approach specifically consisted of:

1. Merging the text of all cluster documents into a single *pseudo-document*.
2. Ranking words according to the *text frequency–inverse document frequency* scheme ("tf-idf"), which assigns importance weights by balancing frequency inside the same document with frequency on the whole collection [17].
3. Selecting the top k words with different lexical stems[3].

For representative document subset generation, *degree centrality* (a social network analysis metric that quantifies the importance of a node in a network) was calculated for every node (document) of the induced subgraph of each cluster; this allowed to rank documents.

An example of topic properties is shown in Table 1.

4 Experiments and Results

To test the proposed approach, we clustered a dataset of the 2005 English Wikipedia (pre-processed with Wikiprep[4]), which approximately consists of 800, 000 content pages (i.e., pages that are not categories or lists) and 19 million links. Because we are seeking for graph clusters, only those groups with $\rho >= 0.5$ were kept; this gave a total of 55,805 document groups.

The aim of validation follows two lines: (1) measuring clustering quality and (2) confirming that the extracted groups correspond to topics. For these purposes, internal and external validation techniques were applied over our results. For the former, we compared intra vs. inter cluster proximity; for the latter, user tests and an alignment with Wikipedia's category network were carried out. For additional information on these experiments (specially for replication purposes), please refer to Garza's thesis [5]. Also, an earlier work by Garza and Brena shows an initial approach and preliminary results [6].

[3] *Stemming* consists of determining the base form of a word; this causes terms like "runner" and "running" to be equivalent, as their base form is the same ("run").

[4] `http://www.cs.technion.ac.il/~gabr/resources/code/wikiprep/`

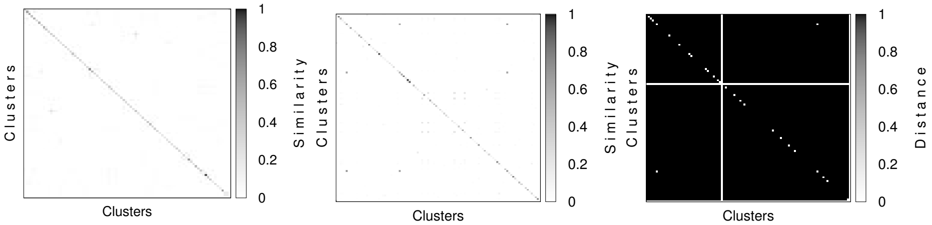
4.1 Internal Validation

For internal validation, we employed visual *proximity matrices*, in which the intensity of each cell indicates the proximity (either similarity or distance) between a pair of clusters (obtained, in our case, by taking the average proximity that results from calculating proximity between pairs of cluster documents). Of course, proximity among elements of the same cluster (represented by the main diagonal) should be greater than proximity among elements of different clusters; consequently, an outstanding main diagonal should be present on the matrix.

Three proximity metrics were used for these tests: cosine similarity, semantic relatedness, and Jaccard similarity. The first (and most important one for our purposes) takes word vectors as input and is thus orthogonal to our clustering method, since we do not employ text (this, in fact, makes the validation *semantic*); the second metric calculates distance specifically for Wikipedia articles and is based on the Google Normalized Distance [15]. The third metric is a standard set-similarity measurement.

For setup, a systematic sample of 100 clusters was chosen (the clusters being sorted by relative density); each cluster was seen in terms of its 30 most representative documents. Regarding cosine similarity, the word vectors were constructed from the cluster documents' text; regarding semantic relatedness and the Jaccard similarity, binary vectors indicating hyperlink presence or absence in documents were constructed.

Figure 1 presents the resulting similarity matrices; as we can see, the main diagonal clearly outstands from the rest of the matrix. Intra-cluster similarity was on average 46 and 190 times higher than inter-cluster similarity for cosine and Jaccard similarity, respectively. For semantic relatedness, the ratio of unrelated articles (infinite distance) was twice higher among elements of different clusters.



(a) Cosine (b) Jaccard (c) Semantic Relatedness

Fig. 1. Resulting similarity matrices. Note that for semantic relatedness low values are favorable, as it consists of a distance (dissimilarity) metric.

4.2 External Validation

We now describe user tests and the alignment with Wikipedia's category network.

User Tests. To test coherence on our topics an *outlier detection* user task (based on Chang's [2]) was designed. On each individual test (one per cluster), users were presented two lists: a *member list* with titles from actual documents of the cluster and a *test list* that mixed true members with outliers. Users were told to correctly detect all of the latter items. To measure quality, standard accuracy measures such as precision, recall, and F_1 were calculated.

200 clusters— represented by their properties —were randomly selected for the test set (outliers were also chosen at random). To prevent careless answers (e.g., selection of all items on the test list), two items from the member list were copied into the test list (tests with these elements marked were discarded). The test set was uploaded to Amazon's Mechanical Turk[5], a reliable on-line platform that hosts tasks to be performed by anonymous workers (users) for a certain fee.

As for results, a 366 tests were answered; 327 of them were valid (89%). F_1 was 0.92 on average (an almost perfect score); for more details see Figure 2b. In that sense, we can argue that users found sense in our topics.

Alignment with Wikipedia's Category Network. The alignment consisted of mapping our clusters to Wikipedia categories (1:1 relationships); from each mapping, standard accuracy measures such as precision, recall, and F_1 were calculated.

Although F_1 was 0.53 on average, more than 20% of the clusters accomplished a perfect or nearly perfect score (most had $\rho \approx 1.0$). Furthermore, a moderate correlation was found between ρ and F_1; this correlation supports our intuitive assumption of structure indicating topicality. Table 1 presents a few clusters with their matching categories, and Figure 2a shows curves for F_1 and precision vs. recall.

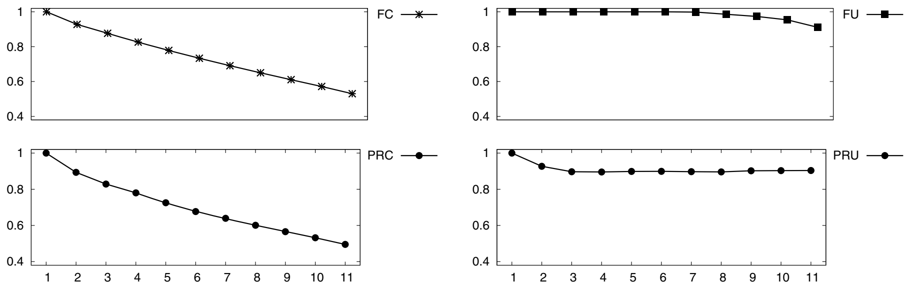
To sum validation up, we can state that all tests provided evidence to support our hypothesis of graph clusters being topics: internal evaluation with cosine similarity not only showed that documents of the same group were textually similar (an indicator of "topicness"); there is a correlation between our structural cohesion function and the score obtained by measuring resemblance with Wikipedia categories, and users found a logical sense to the clusters presented in the outlier detection tests.

5 Related Work

Related work revolves around the three axes mentioned in Section 2: Web structure, topic, and Wikipedia mining. Approaches that intersect at several axes are now discussed.

Topic and Wikipedia mining. Topic modeling by clustering keywords with a distance metric based on the Jensen-Shannon divergence is the main contribution of Wartena and Brussee [22]; this approach was tested over a subset of the Dutch Wikipedia. On the other hand, Schönhofen [19] does topic labeling with the aid

[5] http://www.mturk.com



(a) Category alignment

(b) User tests

Fig. 2. External evaluation results. FC=F-score curve for category alignment tests (scores sorted in descending order), PRC=Precision vs. recall standard 11-level curve for category alignment tests, FU=F-score curve for user tests, and PRU=Precision vs. recall curve for user tests.

Table 1. Aligned clusters

beatles; lennon; mc cartney; song	**artery; vein; anatomy; blood**	**paralympics; summer; winter; games**
Category: The Beatles *Cluster size:* 351 F_1: 0.71 ρ: 0.51	*Category:* Arteries *Cluster size:* 94 F_1 score: 0.62 Rel. density (ρ): 0.9	*Category:* Paralympics *Cluster size:* 32 F_1: 0.92 ρ: 1.0
The Beatles The Beatles discography John Lennon Paul McCartney George Harrison Ringo Starr	Aorta Pulmonary artery Pulmonary vein Venae cavae Superior vena cava Femoral vein	Paralympic Games 2004 Paralympics 1988 Paralympics 1980 Paralympics

of Wikipedia's base of categories; the aim was to assign labels from Wikipedia to a cluster of documents. This pair of initiatives can be clearly differentiated from our approach: they use content instead of structure and follow a distinct topic mining paradigm (modeling and labeling, respectively, while ours does enumeration, distillation, and modeling). Moreover, Schönhofen, uses Wikipedia more as a source of information (we use it both as source and destination).

Topic and Web structure mining. Modha and Spangler [16] present hypertext clustering based on a hybrid similarity metric, a variant of k-means, and the inclusion of properties ("nuggets") into the clustering process. They carry out topic enumeration, labeling, and distillation. He et al. [9] also do enumeration and distillation by clustering webpages with a spectral method and a hybrid similarity metric; the aim was to list representative webpages given a query.

Although these works discover clusters of topically related documents and either refine those clusters or calculate properties as well, they carry out data clustering (we, in contrast, do graph clustering). Moreover, their information source is mixed, as content and structure are both used for clustering.

6 Conclusions and Future Work

Throughout the present work, we found out that a high relative density in vertex groups indicates that these tend to share a common thematic in Wikipedia-like document collections. This was shown on an experimental basis— mainly with the aid of human judgment and a comparison against a set of reference classes (categories) for Wikipedia.

Also, topic bodies being detected with a local clustering approach solely based on structure was initially stated and shown. While not discarding the utility of hybrid methods (e.g. content and structure), we consider this result to be important; in that sense, GLC-based topic mining might be specially helpful if we have collections with small amounts of text (for example, a scientific collaboration network where only article titles are available).

Regarding future work, it may span throughout several areas: (a) modification of the clustering algorithm (e.g. use of different cohesion functions), (b) management of temporal aspects, and (c) development of applications that benefit from the extracted topics. We also intend to compare our results against other methods, e.g. topic modeling approaches.

References

1. Auer, S., Lehmann, J.: What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 503–517. Springer, Heidelberg (2007)
2. Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., Blei, D.M.: Reading tea leaves: How humans interpret topic models. In: Neural Information Processing Systems (2009)
3. Chen, J., Zaiane, O.R., Goebel, R.: Detecting Communities in Large Networks by Iterative Local Expansion. In: International Conference on Computational Aspects of Social Networks 2009, pp. 105–112. IEEE (2009)
4. Flake, G.W., Lawrence, S., Giles, C.L.: Efficient identification of Web communities. In: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 150–160. ACM, New York (2000)
5. Garza, S.E.: A Process for Extracting Groups of Thematically Related Documents in Encyclopedic Knowledge Web Collections by Means of a Pure Hyperlink-based Clustering Approach. PhD thesis, Instituto Tecnológico y de Estudios Superiores de Monterrey (2010)
6. Garza, S.E., Brena, R.F.: Graph Local Clustering for Topic Detection in Web Collections. In: 2009 Latin American Web Congress, pp. 207–213. IEEE (2009)
7. Gibson, D., Kumar, R., Tomkins, A.: Discovering large dense subgraphs in massive graphs. In: Proceedings of the 31st International Conference on Very Large Data Bases, pp. 721–732. VLDB Endowment (2005)

8. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Science USA 101(1), 5228–5235 (2004)
9. He, X., Ding, C.H.Q., Zha, H., Simon, H.D.: Automatic topic identification using webpage clustering. In: Proceedings of the IEEE International Conference on Data Mining, pp. 195–202 (2001)
10. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
11. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics 11, 33015 (2009)
12. Liu, Y., Niculescu-Mizil, A., Gryc, W.: Topic-link LDA: joint models of topic and author community. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, New York (2009)
13. Luo, F., Wang, J.Z., Promislow, E.: Exploring local community structures in large networks. Web Intelligence and Agent Systems 6(4), 387–400 (2008)
14. Menczer, F.: Links tell us about lexical and semantic web content. CoRR, cs.IR/0108004 (2001)
15. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 509–518. ACM, New York (2008)
16. Modha, D.S., Spangler, W.S.: Clustering hypertext with applications to Web searching, US Patent App. 10/660,242 (September 11, 2003)
17. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management, 513–523 (1988)
18. Schaeffer, S.E.: Stochastic Local Clustering for Massive Graphs. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 354–360. Springer, Heidelberg (2005)
19. Schönhofen, P.: Identifying document topics using the Wikipedia category network. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 456–462. IEEE Computer Society, Washington, DC, USA (2006)
20. Stein, B., Zu Eissen, S.M.: Topic identification: Framework and application. In: Proceedings of the International Conference on Knowledge Management, vol. 399, pp. 522–531 (2004)
21. Virtanen, S.E.: Clustering the Chilean Web. In: Proceedings of the 2003 First Latin American Web Congress, pp. 229–231 (2003)
22. Wartena, C., Brussee, R.: Topic detection by clustering keywords. In: DEXA 2008: 19th International Conference on Database and Expert Systems Applications (2008)

SC Spectra: A Linear-Time Soft Cardinality Approximation for Text Comparison

Sergio Jiménez Vargas[1] and Alexander Gelbukh[2]

[1] Intelligent Systems Research Laboratory (LISI),
Systems and Industrial Engineering Department
National University of Colombia, Bogota, Colombia
sgjimenezv@unal.edu.co

[2] Center for Computing Research (CIC)
National Polytechnic Institute (IPN), Mexico City, Mexico
www.gelbukh.com

Abstract. Soft cardinality (SC) is a softened version of the classical cardinality of set theory. However, given its prohibitive cost of computing (exponential order), an approximation that is quadratic in the number of terms in the text has been proposed in the past. SC Spectra is a new method of approximation in linear time for text strings, which divides text strings into consecutive substrings (i.e., q-grams) of different sizes. Thus, SC in combination with resemblance coefficients allowed the construction of a family of similarity functions for text comparison. These similarity measures have been used in the past to address a problem of entity resolution (name matching) outperforming SoftTFIDF measure. SC spectra method improves the previous results using less time and obtaining better performance. This allows the new method to be used with relatively large documents such as those included in classic information retrieval collections. SC spectra method exceeded SoftTFIDF and cosine tf-idf baselines with an approach that requires no term weighing.

Keywords: approximate text comparison, soft cardinality, soft cardinality spectra, q-grams, ngrams.

1 Introduction

Assessment of similarity is the ability to balance both commonalities and differences between two objects to produce a judgment result. People and most animals have this intrinsic ability, making of this an important requirement for artificial intelligence systems. Those systems rarely interact with objects in real life, but they do with their data representations such as texts, images, signals, etc. The exact comparison of any pair of representations is straightforward, but unlike this crisp approach, the approximate comparison has to deal with noise, ambiguity and implicit information, among other issues. Therefore, a challenge for many artificial intelligence systems is that their assessment of the similarity be, to some degree, in accordance with human judgments.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 213–224, 2011.

For instance, names are the text representation–sometimes quite complex, cf. [3,2]–most commonly used to refer to objects in real life. Like humans, intelligent systems when referring to names have to deal with misspellings, homonyms, initialisms, aliases, typos, and other issues. This problem has been studied by different scientific communities under different names, including: record linkage [23], entity resolution [12], object identification [22] and (many) others.

The name matching task [4] consists of finding co-referential names in a pair of lists of names, or to find duplicates in a single list. The methods that use pairs of surface representations are known as static methods and usually tackle the problem using a binary similarity function and a decision threshold. On the other hand, adaptive approaches make use of information throughout the list of names. The adaptability of several of these approaches usually relies on the *tf-idf* weighting or similar methods [20].

Comparison methods can also be classified by the level of granularity in which the texts are divided. For example, the family of methods derived from the edit distance [15] use characters as a unit of comparison. The granularity is increased gradually in the methods based on q-grams of characters [13]. Q-grams are consecutive substrings of length q overlapping $q-1$ characters, also known as *kmers* or *ngrams*. Further, methods such as vector space model (VSM) [20] and coefficients of similarity [21] make use of terms (i.e., words or symbols) as sub-division unit. The methods that have achieved the best performance in the entity resolution task (ER) are those that combine term-level comparisons with comparisons at character or q-gram level. Some examples of these hybrid approaches are Monge-Elkan's measure [17,10], SoftTFIDF [8], fuzzy match similarity (FMS) [5], meta-levenshtein (ML) [18] and soft cardinality (SC) [11].

Soft cardinality is a set-based method for comparing objects that softens the crisp counting of elements that makes the classic set cardinality, considering the similarities among elements. For text comparisons, the texts are represented as sets of terms. The definition of SC requires the calculation of 2^m intersections for a set with m terms. Jimenez *et al.* [11] proposed an approach to SC using only m^2 computations of an auxiliary similarity measure that compares two terms.

In this paper, we propose a new method of approximation for SC that unlike the current approach does not require any auxiliary similarity measure. In addition, the new method allows simultaneous comparison of uni-grams (i.e., characters), bi-grams or tri-grams by combining a range of them. We call these combinations SC spectra (soft cardinality spectra). SC spectra can be computed in linear time allowing the use of soft cardinality with large texts and in other intelligent-text-processing applications such as information retrieval. We tested SC spectra with 12 entity resolution data sets and with 9 classic information retrieval collections overcoming baselines and the previous SC approximation.

The remainder of this paper is organized as follows: Section 2 briefly recapitulates the SC method for text comparison. The proposed method is presented in Section 3. In Section 4, the proposed method is experimentally compared with the previous approximation method and other static and adaptive approaches; a brief discussion is provided. Related work is presented in Section 5. Finally, in Section 6 conclusions are given and future work is briefly discussed.

2 Soft Cardinality for Text Comparison

The cardinality of a set is defined as the number of different elements in itself. When a text is represented as a bag of words, the cardinality of the bag is the size of its vocabulary of terms. Rational cardinality-based similarity measures are binary functions that compare two sets using only the cardinality of each set and - at least - the cardinality of their union or intersection. Examples of these measures are *Jaccard* ($|A \cap B|/|A \cup B)$, *Dice* ($2|A \cap B|/(|A| + |B|)$) and *cosine* ($|A \cap B|/\sqrt{|A||B|}$) coefficients. The effect of the cardinality function in these measures is to count the number of common elements and compressing repeated elements in a single instance. On the basis of an information theoretical definition of similarity proposed by Lin [16], Cilibrasi and Vitányi [7] proposed a compression distance that takes advantage of this feature explicitly showing its usefulness in text applications.

However, the compression provided by classical cardinality is crisp. That is, two identical elements in a set are counted once, but two nearly identical elements count twice. This problem is usually addressed in text applications using *stemming*, but this approach is clearly not appropriate for name matching. Soft cardinality (SC) addresses this issue taking into account the similarities between elements of the set. SC's intuition is as follows: the elements that have similarities with other elements contribute less to the total cardinality than unique elements.

2.1 Soft Cardinality Definition

The soft cardinality of a set is the cardinality of the union of its elements treated themselves as sets. Thus, for a set $A = \{a_1, a_2, \ldots, a_n\}$, the soft cardinality of A is $|A|'=|\bigcup_{i=1}^{n} a_i|$.

Representing text as bag of words, two names such as “Sergio Gonzalo Jiménez” and “Cergio G. Gimenes” can be divided into terms (tokens) and compared using soft cardinality as it is depicted in Fig. 1. Similarities among terms are represented as intersections. The soft cardinality of each set is represented as the area inside of the resulting cloud-border shape. Similarity measures can be obtained using resemblance coefficients, such as *Jaccard*, obtaining: $sim(A, B) = (|A|' + |B|' - |A \cup B|')/|A \cup B|'|$.

2.2 SC Approximation with Similarity Functions

Computing cardinality of the union of n sets requires the addition of $2^n - 1$ numbers. Besides, each one of those values can be the intersection of n sets. For

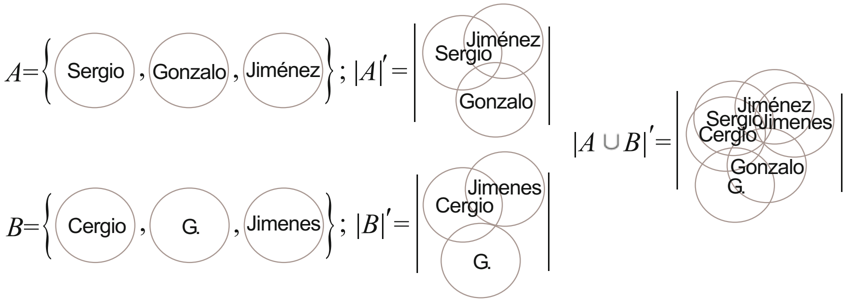


Fig. 1. Example

instance, the cardinality of the union of three sets is $|r \cup s \cup t| = |r| + |s| + |t| - |r \cap s| - |s \cap t| - |r \cap t| + |r \cap s \cap t|$. Even for small values of n this computation is not practical.

The soft cardinality can be approximated by using only pairwise comparisons of elements with the following expression:

$$|A|'_\alpha \simeq \sum_i^n \left(\sum_j^n \alpha(a_i, a_j)^p \right)^{-1} \tag{1}$$

This approximation method makes n^2 calculations of the similarity function $\alpha(*, *)$, which has range $[0, 1]$ and satisfies $\alpha(x, x) = 1$. In our scenario, this function returns the similarity between two terms. In fact, when α is a crisp comparator (i.e., returns 1 when the elements are identical and 0 otherwise) $|A|'_\alpha$ becomes $|A|$, i.e., the classical set cardinality. Finally, the exponent p is a tuning parameter investigated by Jimenez *et al.* [11], who obtained good results using $p = 2.0$ in a name-matching task.

3 Computing Soft Cardinality Using Sub-strings

The SC approximation shown in (1) is quite general since the function of similarity between the terms α may or may not use the surface representation of both strings. For example, the edit distance is based on a surface representation of characters, in contrast to a semantic relationship function, which can be based on a large corpus or a semantic network. Furthermore, when the surface representation is being used, SC could be calculated by subdividing the text string into substrings and then count the number of different substrings. However, if the unit of the subdivision is q-grams of characters, the resulting similarity measure would ignore the natural subdivision in terms (tokens) of the text string.

Several comparative studies have shown the convenience of the hybrid approaches that first tokenize (split in terms) a text string and then make comparisons between the terms at character or q-gram level [8,4,6,19,11]. Similarly, the definition of SC is based on an initial tokenization and an implicit further subdivision made by the function α to assess similarities and differences between pairs of terms. The intuition behind the new SC approximation is first tokenizing the text. Second, to split each term into a finer-grained substring unit (e.g., bi-grams). Third, to make a list of all the different substrings, and finally, calculate a weighted sum of the sub-strings with weights that depends on the number of substrings in each term.

Consider the following example with the Spanish name "Gonzalo Gonzalez", A ={"Gonzalo","Gonzalez"}, a_1 ="Gonzalo" and a_2 ="Gonzalez". Using bi-grams with padding characters[1] as subdivision unit; the pair of terms can be represented as: $a_1^{[2]} = \{\triangleleft G, Go, on, nz, za, al, lo, o\triangleright\}$ and $a_2^{[2]} = \{\triangleleft G, Go, on, nz, za, al, le, ez, z\triangleright\}$. The exponent in square brackets means the size q of the q-gram subdivision. Let $A^{[2]}$ be the set with all different bi-grams $A^{[2]} = a_1^{[2]} \cup a_2^{[2]} = \{\triangleleft G, Go, on, nz, za, al, lo, o\triangleright, le, ez, z\triangleright\}$, $|A^{[2]}| = |a_1^{[2]} \cup a_2^{[2]}| = 11$. Similarly, $|a_1^{[2]} - a_2^{[2]}| = 2$, $|a_2^{[2]} - a_1^{[2]}| = 3$ and $|a_1^{[2]} \cap a_2^{[2]}| = 6$.

Thus, each one of the elements of $A^{[2]}$ adds a contribution to the total soft cardinality of A. The elements of $A^{[2]}$ that also belongs to $a_1^{[2]} - a_2^{[2]}$ or $a_2^{[2]} - a_1^{[2]}$ contributes $1/|a_1^{[2]}| = 0.125$ and $1/|a_2^{[2]}| = 0.11\bar{1}$ respectively; that is the inverse of the number of bi-grams on each term. Common bi-grams between $a_1^{[2]}$ and $a_2^{[2]}$ must contribute with a value in $[0.11\bar{1}, 0.125]$ interval. The most natural choice, given the geometrical metaphor depicted in Fig. 1, is to select the maximum. Finally, soft cardinality for this example is $|A|' \simeq 0.125\times 2 + 0.11\bar{1}\times 3 + 0.125\times 6 = 1.33\bar{3}$ in contrast to $|A| = 2$. The soft cardinality of A reflects the fact that a_1 and a_2 are similar.

3.1 Soft Cardinality q-Spectrum

The SC of a text string can be approximated using a partition $A^{[q]} = \bigcup_{i=1}^{|A|} a_i^{[q]}$ of A in q-grams, where $a_i^{[q]}$ is the partition of i-th term in q-grams. Clearly, each one of the q-grams $A_j^{[q]}$ in $A^{[q]}$ can occur in several terms a_i of A, having indices i satisfying $A_j^{[q]} \in a_i^{[q]}$. The contribution of $A_j^{[q]}$ to the total SC is the maximum of $1/|a_i^{[q]}|$ for each one of its occurrences. The final expression for SC is:

$$|A|'_{[q]} \simeq \sum_{j=1}^{|A^{[q]}|} \max_{i; A_j^{[q]} \in a_i^{[q]}} \left(\frac{1}{|a_i^{[q]}|} \right). \qquad (2)$$

The approximation $|A|'_{[q]}$ obtained with (2) using q-grams is the SC q-spectrum of A.

[1] Padding characters are especial characters padded at the begining and the end of each term before being subdivided in q-grams. These characters allows to distinguish heading and trailing q-grams from those at the middle of the term.

3.2 Soft Cardinality Spectra

A partition of q-grams allows the construction of similarity measures with its SC q-spectrum associated. The most fine-grained subtring partition is $q = 1$ (i.e., characters) and the coarser is the partition into terms. While partitions such as uni-grams, bi-grams and tri-grams are used in tasks such as entity resolution, the term partition is preferred for information retrieval, text classification and others. Intuitively, finer partitions appear to be suitable for short texts -such as names- and terms seem to be more convenient for documents.

The combination of several contiguous partition granularities can be useful for comparing texts in a particular dataset. Given that each SC q-spectrum provides a measure of the compressed amount of terms in a text, several SC q-spectrum can be averaged or added to get a more meaningful measure. SC spectra is defined as the addition of a range of q-spectrum starting at q_s and ending at q_e, denoted SC spectra $[q_s : q_e]$, having $q_s \leq q_e$. For instance, the SC spectra $[2 : 4]$ uses simultaneously bi-grams, tri-grams and quad-grams to approximate the soft cardinality of a bag of words. Thus, the SC spectra expression is:

$$|A|'_{[q_s:q_e]} = \sum_{i=s}^{e} |A|'_{[q_i]}. \tag{3}$$

4 Experimental Evaluation

The proposed experimental evaluation aims to address the following issues: (i) to determine which of the different substring padding approaches are more suitable for entity resolution (ER) and information retrieval (IR) tasks, (ii) to determine if SC spectra is more convenient than SC q-spectrum, (iii) to compare SC spectra versus the previous SC approximation, (iv) to compare the performance of the proposed similarity measure obtained using SC spectra versus other text measures.

4.1 Experimental Setup

Data Sets. For experimental evaluation, two groups of data sets were used for entity resolution and information retrieval tasks, respectively. The first group, called ER, consists of twelve data sets for name matching collected from different sources under secondstring framework[2]. The second group, called IR, is composed of nine information retrieval classic collections described by Baeza-Yates and Ribeiro-Neto [1][3]. Each data set is composed of two sets of texts and a gold-standard relation that associates pairs from both sets. The gold-standard in all data sets was obtained from human judgments, excluding *census* and *animal* data sets that were built, respectively, making random edit operations into a list of people names, and using a single list of animal names and considering as co-referent names pairs who are proper sets at term level. At ER data sets,

[2] http://secondstring.sourceforge.net/

[3] http://people.ischool.berkeley.edu/~hearst/irbook/

gold-standard relationship means identity equivalence, and at IR data sets, it means relevance between a query or information need and a document.

Texts in all data sets were divided into terms—i.e., tokenized—with a simple approach using as separator the space character, punctuation, parenthesis and others special characters such as slash, hyphen, currency, tab, etc. Besides, no stop words removal or stemming was used.

Text Similarity Function. The text similarity function used to compare strings was built using a cardinality-based resemblance coefficient replacing classic set cardinality by SC spectra. The used resemblance coefficient was the quotient of the cardinality of intersection divided by the harmonic mean of individual cardinalities:

$$harmonic(A,B) = \frac{|A \cap B| \times (|A| + |B|)}{2 \times |A| \times |B|}. \tag{4}$$

The intersection operation in (4) can be replaced by union using $|A \cap B| = |A| + |B| - |A \cup B|$. Thus, the final text similarity function between two tokenized text strings A and B is given by the following expression:

$$sim(A,B) = 1 + \frac{1}{2}\left(\frac{|A|'_{[q_s:q_e]}}{|B|'_{[q_s:q_e]}} + \frac{|B|'_{[q_s:q_e]}}{|A|'_{[q_s:q_e]}} - \frac{|A \cup B|'_{[q_s:q_e]}}{|A|'_{[q_s:q_e]}} - \frac{|A \cup B|'_{[q_s:q_e]}}{|B|'_{[q_s:q_e]}} \right). \tag{5}$$

Performance Measure. The quality of the similarity function proposed in (5) can be quantitatively measured using several performance metrics for ER and IR tasks. We preferred to use interpolated average precision (IAP) because is a performance measure that has been commonly used at both tasks (see [1] for a detailed description). IAP is the area under precision-recall curve interpolated at 11 evenly separated recall points.

Experiments. For experiments, 55 similarity functions were constructed with all possible SC spectra using q-spectrum ranging q from 1 to 10 in combination with (5). Each obtained similarity function was evaluated using all text pairs into the entire Cartesian product between both text sets on all 19 data set. Besides, three padding approaches were tested:

single padding to pad one character before and after each token, e.g. the [2:3] spectra sub-division of "sun" is $\{\triangleleft s, su, un, n\triangleright, \triangleleft su, sun, un\triangleright\}$.

full padding to pad $q-1$ characters before and after each token, e.g. the [2:3] spectra sub-division of "sun" is $\{\triangleleft s, su, un, n\triangleright, \triangleleft\triangleleft s, \triangleleft su, sun, un\triangleright, n\triangleright\triangleright\}$.

no padding e.g.[2:3] spectra for "sun" is $\{su, un, sun\}$

For each one of the 3135 (55×19×3) experiments carried out interpolated average precision was computed. Fig. 2 shows a results sample for two data sets—*hotels* and *adi*—using *single padding* and *no padding* configurations respectively.

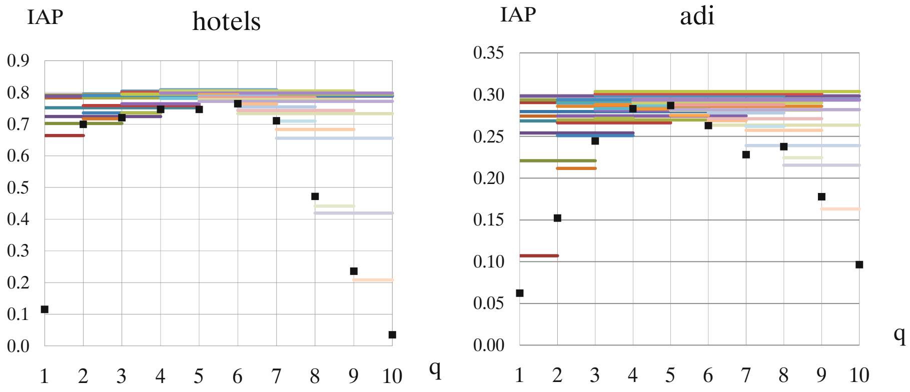


Fig. 2. IAP performance for all SC spectra form $q = 1$ to $q = 10$ for data sets *hotels* and *adi*. Spectra with single q-spectrum are shown as black squares (e.g. [3:3]). Wider spectra are shown as horizontal bars.

4.2 Results

Tables 1 and 2 show the best SC spectra for each data set using the three proposed padding approaches. *Single padding* and *no padding* seems to be more convenient for ER and IR data set groups respectively.

Table 1. Results for best SC spectra using ER data sets

PADDING	full		single		no	
DATA SET	spectra	IAP	spectra	IAP	spectra	IAP
birds-scott1	**[1:2]***	**0.9091**	**[1:2]***	**0.9091**	**[1:2]***	**0.9091**
birds-scott2	[7:8]*	0.9005	**[6:10]**	**0.9027**	[5:9]	0.9007
birds-kunkel	[5:7]*	0.8804	**[6:6]**	**0.8995**	[4:4]	0.8947
birds-nybird	[4:6]	0.7746	**[1:7]**	**0.7850**	[4:5]	0.7528
business	[1:3]	0.7812	**[1:4]**	**0.7879**	[1:4]	0.7846
demos	**[2:2]**	**0.8514**	**[2:2]**	**0.8514**	[1:3]	0.8468
parks	[2:2]	0.8823	[1:9]	0.8879	**[2:4]**	**0.8911**
restaurant	[1:6]	0.9056	**[3:7]**	**0.9074**	**[1:6]**	**0.9074**
ucd-people	**[1:2]***	**0.9091**	**[1:2]***	**0.9091**	**[1:2]***	**0.9091**
animal	[1:10]	0.1186	**[3:8]**	**0.1190**	[3:4]	0.1178
hotels	[3:4]	0.7279	[4:7]	0.8083	**[2:5]**	**0.8147**
census	[2:2]	0.8045	**[1:2]**	**0.8110**	[1:2]	0.7642
best average	[3:3]	0.7801	**[2:3]**	**0.7788**	[1:3]	0.7746
average of best		0.7871		**0.7982**		0.7911

* Asterisks indicate that another wider SC spectra also showed the same IAP performance.

Table 2. Results for best SC spectra using IR collections

PADDING	full		single		no	
DATA SET	spectra	IAP	spectra	IAP	spectra	IAP
cran	**[7:9]**	**0.0070**	[3:4]	0.0064	[3:3]	0.0051
med	[4:5]	0.2939	**[5:7]***	**0.3735**	[4:6]	0.3553
cacm	**[4:5]**	**0.1337**	[2:5]	0.1312	[2:4]	0.1268
cisi	[1:10]	0.1368	[5:8]	0.1544	**[5:5]**	**0.1573**
adi	[3:4]	0.2140	[5:10]	0.2913	**[3:10]**	**0.3037**
lisa	[3:5]	0.1052	[5:8]	0.1244	**[4:6]**	**0.1266**
npl	[7:8]	0.0756	[3:10]	0.1529	**[3:6]**	**0.1547**
time	[1:1]	0.0077	[8:8]	0.0080	**[6:10]**	**0.0091**
cf	[7:9]	0.1574	[5:10]	0.1986	**[4:5]**	**0.2044**
best average	[3:4]	0.1180	**[5:8]**	**0.1563**	[4:5]	0.1542
average of best		0.1257		0.1601		**0.1603**

* Asterisks indicate that another wider SC spectra also showed the same IAP performance.

Fig. 3 shows precision-recall curves for SC spectra in comparison with other measures. The series named *best SC spectra* is the average of the best SC spectra for each data set using *single padding* for ER and *no padding* for IR. *MongeElkan* measure [17] used an internal inter-term similarity function of bi-grams combined with Jaccard coefficient. *SoftTFIDF* used the same configuration proposed by Cohen *et al.* [8] but fixing its normalization problem found by Moreau *et al.* [18]. Soft Cardinality used ([1]) with $p = 2$ and the same inter-term similarity function used with *MongeElkan* measure.

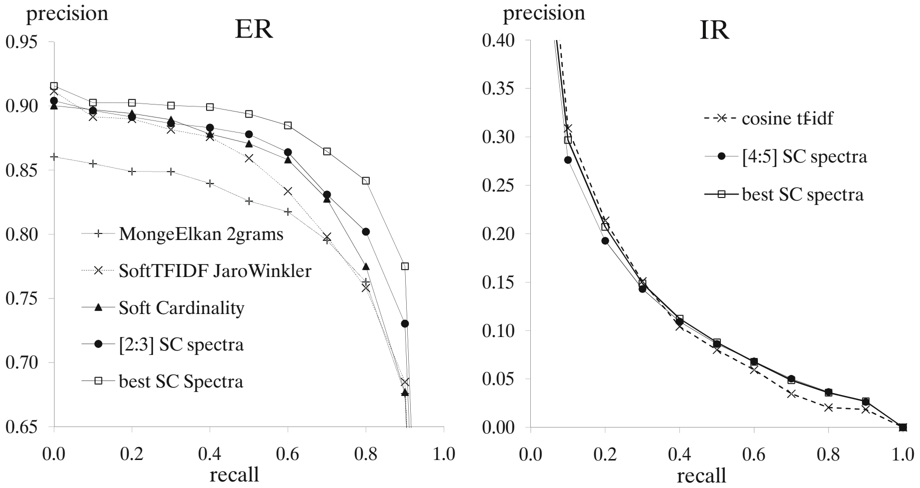


Fig. 3. Precision-recall curves of SC spectra and other measures

4.3 Discussion

Results in Tables 1 and 2 indicate that padding characters seem to be more useful at ER data sets than at IR collections, but using only a single padding character. Apparently, the effect of adding padding characters is important only in collections with relatively short texts such as ER.

Best performing configurations (showed in boldface) were reached—in most of the cases (16 over 19)—using SC spectra instead of single SC q-spectrum. This effect can also be appreciated in Figures 2 (a) and (b), where SC spectra (represented as horizontal bars) tends to outperform SC q-spectrum (represented as small black squares). The relative average improvement of the best SC spectra for each data set versus the best SC q-spectrum was 1.33% for ER data sets and 4.48% for IR collections. Results for best SC q-spectrum were not shown for space limitations. In addition, Fig. 2 qualitatively shows that SC spectra measures tend to perform better than the SC q-spectrum with maximum performance of those that compose a SC spectra. For instance, [7:9] SC spectra at *adi* collection outperforms all SC 7-grams, SC 8-grams and SC 9-grams.

As Fig. 3 clearly shows—for ER data—the similarity measures obtained using the best SC spectra for each data set outperforms the other tested measures. It is important to note that unlike SoftTFIDF, measures obtained using SC spectra are static. That is, they do not use term weighting obtained from term frequencies into the entire data set. Regarding IR, SC spectra reached practically the same performance than *cosine tf-idf*. This result is also remarkable because we are reaching equivalent performance (better at ER data) using considerably less information. Finally, ER results also show that SC spectra is a better soft cardinality approximation than the previous approximation; see (1). Besides, SC spectra require considerably less computational effort than that approximation.

5 Related Work

The proposed weighting schema that gives smaller weights to substrings according to the length in characters of each term is similar to the approach of De La Higuera & Micó, who assigned a variable cost to character edit operations to Levenshtein's edit distance [9]. They obtained improved results in a text classification task using this cost weighting approach. This approach is equivalent to ours because the contribution of each q-gram to the SC depends on the total number of q-grams in the term, which in turn depends on the length in characters of the term.

Leslie *et al.* [14] proposed a k-spectrum kernel for comparing sequences using sub-strings of k-length in a protein classification task. Similarly to them, we use the same metaphor to name our approach.

6 Conclusions and Future Work

We found that the proposed SC spectra method for text comparison performs particularly well for the entity resolution problem and reach the same results of cosine *tf-idf* similarity using classic information retrieval collections. Unlike several current approaches, SC spectra does not require term weighting. However, as future work, it is interesting to investigate the effect of weighting in SC spectra at term and substring level. Similarly, how to determine the best SC spectra for a particular data set is an open question worth to investigate. Finally, we also found that SC spectra is an approximation for soft cardinality with less computational cost and better performance, allowing the proposed method to be used with longer documents such as those of text information retrieval applications.

Acknowledgements. This research was funded in part by the Systems and Industrial Engineering Department, the Office of Student Wellfare of the National University of Colombia, Bogotá, and throught a grant from the Colombian Department for Science, Technology and Innovation Colciencias, project 110152128465. The second author recognizes the support from Mexican Government (SNI, COFAA-IPN, SIP 20113295, CONACYT 50206-H) and CONACYT–DST India (project "Answer Validation through Textual Entailment").

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley & ACM Press (1999)
2. Barceló, G., Cendejas, E., Bolshakov, I., Sidorov, G.: Ambigüedad en nombres hispanos. Revista Signos. Estudios de Lingüística 42(70), 153–169 (2009)
3. Barceló, G., Cendejas, E., Sidorov, G., Bolshakov, I.A.: Formal Grammar for Hispanic Named Entities Analysis. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 183–194. Springer, Heidelberg (2009)
4. Bilenko, M., Mooney, R., Cohen, W.W., Ravikumar, P., Fienberg, S.: Adaptive name matching in information integration. IEEE Intelligent Systems 18(5), 16–23 (2003), http://portal.acm.org/citation.cfm?id=1137237.1137369
5. Chaudhuri, S., Ganjam, K., Ganti, V., Motwani, R.: Robust and efficient fuzzy match for online data cleaning. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 313–324. ACM, San Diego (2003), http://portal.acm.org/citation.cfm?id=872757.872796
6. Christen, P.: A comparison of personal name matching: Techniques and practical issues. In: International Conference on Data Mining Workshops, pp. 290–294. IEEE Computer Society, Los Alamitos (2006)
7. Cilibrasi, R., Vitanyi, P.: Clustering by compression. IEEE Transactions on Information Theory, 1523–1545 (2005)
8. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: Proceedings of the IJCAI 2003 Workshop on Information Integration on the Web, pp. 73–78 (August 2003), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.15.178
9. de la Higuera, C., Mico, L.: A contextual normalised edit distance. In: IEEE 24th International Conference on Data Engineering Workshop, Cancun, Mexico, pp. 354–361 (2008), http://portal.acm.org/citation.cfm?id=1547551.1547758
10. Jimenez, S., Becerra, C., Gelbukh, A., Gonzalez, F.: Generalized Mongue-Elkan Method For Approximate Text String Comparison. In: Gelbukh, A. (ed.) CICLing 2009. LNCS, vol. 5449, pp. 559–570. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-00382-0_45

11. Jimenez, S., Gonzalez, F., Gelbukh, A.: Text Comparison Using Soft Cardinality. In: Chavez, E., Lonardi, S. (eds.) SPIRE 2010. LNCS, vol. 6393, pp. 297–302. Springer, Heidelberg (2010), http://www.springerlink.com/content/x1w783135m36k880/
12. Köpcke, H., Thor, A., Rahm, E.: Evaluation of entity resolution approaches on real-world match problems. In: Proceedings of the 36th International Conference on Very Large Data Bases, Singapore (2010)
13. Kukich, K.: Techniques for automatically correcting words in text. ACM Computing Surveys 24, 377–439 (1992)
14. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: Biocomputing 2002 - Proceedings of the Pacific Symposium, Kauai, Hawaii, USA, pp. 564–575 (2001), http://eproceedings.worldscinet.com/9789812799623/9789812799623_0053.html
15. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
16. Lin, D.: Information-Theoretic definition of similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 296–304 (1998), http://portal.acm.org/citation.cfm?id=645527.657297&coll=Portal&dl=GUIDE&CFID=92419400&CFTOKEN=72654004
17. Monge, A.E., Elkan, C.: The field matching problem: Algorithms and applications. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, pp. 267–270 (August 1996)
18. Moreau, E., Yvon, F., Cappé, O.: Robust similarity measures for named entities matching. In: Proceedings of the 22nd International Conference on Computational Linguistics, pp. 593–600 (2008), http://portal.acm.org/citation.cfm?id=1599081.1599156
19. Piskorski, J., Sydow, M.: Usability of string distance metrics for name matching tasks in polish. In: Proceedings of the 3rd Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2007), Poznań, Poland, October 5-7 (2007), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.9942
20. Salton, G.: Introduction to modern information retrieval. McGraw-Hill (1983)
21. Sarker, B.R.: The resemblance coefficients in group technology: A survey and comparative study of relational metrics. Computers & Industrial Engineering 30(1), 103–116 (1996), http://dx.doi.org/10.1016/0360-83529500024-0
22. Tejada, S., Knoblock, C.A.: Learning domain independent string transformation weights for high accuracy object identification. In: Proceedings of International Conference on Knowledge Discovery and Data Mining, SIGKDD (2002)
23. Winkler, W.E.: The state of record linkage and current research problems. Statistical research divison U.S. Census Bureau (1999), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.4336

Times Series Discretization Using Evolutionary Programming

Fernando Rechy-Ramírez[1], Héctor-Gabriel Acosta Mesa[1],
Efrén Mezura-Montes[2], and Nicandro Cruz-Ramírez[1]

[1] Departamento de Inteligencia Artificial, Universidad Veracruzana
Sebastián Camacho 5, Centro, Xalapa, Veracruz, 91000, Mexico
frechyr@hotmail.com, {heacosta,ncruz}@uv.mx

[2] Laboratorio Nacional de Informática Avanzada (LANIA) A.C.
Rébsamen 80, Centro, Xalapa, Veracruz, 91000, Mexico
emezura@lania.mx

Abstract. In this work, we present a novel algorithm for time series discretization. Our approach includes the optimization of the word size and the alphabet as one parameter. Using evolutionary programming, the search for a good discretization scheme is guided by a cost function which considers three criteria: the entropy regarding the classification, the complexity measured as the number of different strings needed to represent the complete data set, and the compression rate assessed as the length of the discrete representation. Our proposal is compared with some of the most representative algorithms found in the specialized literature, tested in a well-known benchmark of time series data sets. The statistical analysis of the classification accuracy shows that the overall performance of our algorithm is highly competitive.

Keywords: Times series, Discretization, Evolutionary Algorithms, Optimization.

1 Introduction

Many real-world applications related with information processing generate temporal data [12]. Most of the cases, this kind of data requires huge data storage. Therefore, it is desirable to compress this information maintaining the most important features. Many approaches are mainly focused in data compression. However they do not rely on significant information measured with entropy [11,13]. In those approaches, the dimensionality reduction is given by the transformation of time series of length N into a data set of n coefficients, where $n < N$ [7]. The two main characteristics of a time series are: the number of segments (word size) and the number of values (alphabet) required to represent its continuous values. Fig. 1 shows a time series with a grid that represents the cut points for word size and alphabet.

Most of the discretization algorithms require, as an input, the parameters of word size and alphabet. However, in real-world applications it might be very

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 225–234, 2011.

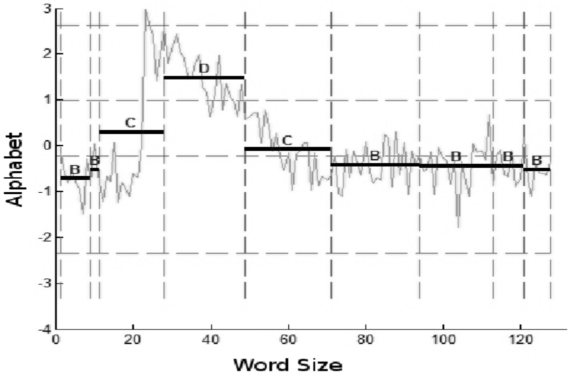


Fig. 1. Word size and alphabet representation. In this case the time series has a word size = 9 and an alphabet = 5 (values A and E do not appear in the time series).

difficult to know in advance their best values. Hence, their definitions require a careful analysis of the time series data set [9,13].

Among the approaches proposed to deal with data discretization we can find those which work with one time series at a time, such as the one proposed by Mörchen [14]. His algorithm is centered on the search of persistent states (the most frequent values) in time series. However, such states are not common in many real-world time series applications. Another representative approach was proposed by Dimitrova [3], where a multi-connected graph representation for time series was employed. The links between nodes have Euclidean distance values which are used under this representation to eliminate links in order to obtain a path that defines the discretization scheme. Nonetheless, this way to define the discretization process could be a disadvantage because not all the time series in a data set will necessarily have the same discretization scheme.

Keogh [13] proposed the *Symbolic Aggregate Approximation (SAX)* approach. This algorithm is based in the *Piecewise Aggregate Approximation (PAA)*, a dimensionality reduction algorithm [8]. After PAA is applied, the values are then transformed into categorical values through a probability distribution function. The algorithm requires the alphabet and the word size as inputs. This is SAX's main disadvantage because it is not clear how to define them from a given time series data set.

There are other approaches based on search algorithms. García-López [2] proposed EBLA2, which in order to automatically find the word size and alphabet performs a greedy search looking for entropy minimization. The main disadvantage of this approach is the sensitivity of the greedy search to get trapped in local optima. Therefore, in [6] simulated annealing was used as a search algorithm and the results improved. Finally, in [1], a genetic algorithm was used to guide the search, however the solution was incomplete in the sense that the algorithm considered the minimization of the alphabet as a first stage, and attempted to reduce the word size in a second stage. In this way some solutions could not be generated.

In this work, we present a new approach in which both, the word size and the alphabet are optimized at the same time. Due to its simplicity with respect to other evolutionary algorithms, evolutionary programming (EP) is adopted as a search algorithm (e.g., no recombination and parent selection mechanisms are performed and just mutation and replacement need to be designed). Furthermore, the amount of strings and the length of the discretized series are optimized as well.

The contents of this paper are organized as follows: Section 2 details the proposed algorithm. After that, Section 3 presents the obtained results and a comparison against other approaches. Finally, Section 4, draws some conclusions and presents the future work.

2 Our Approach

In this section we firstly define the discretization problem. Thereafter, EP is introduced and its adaptation to solve the problem of interest is detailed in four steps: (1) solution encoding, (2) fitness function definition, (3) mutation operator and (4) replacement technique.

2.1 Statement of the Problem

The discretization process refers to the transformation of continuous values into discrete values. Formally, the domain is represented as $x|x \in \mathbf{R}$ where $\mathbf{R}$ is the set of real numbers and the discretization scheme is $D = \{[d_0, d_1], (d_1, d_2], ...$ $(d_{n-1}, d_n]\}$ where d_0 y d_n are the minimum and maximum values for x respectively. Each pair in D represents an interval, where each continuous value is mapped within the continuous values to one of the elements from the discrete set $1...m$, where m is called the discretization degree and $d_i|i = 1...n$ are the limits of intervals, also known as cut points. The discretization process has to be done in both characteristics, the length (word size) and the interval of values taken by the continuous variable (alphabet).

Within our approach we use a modified version of the *PAA* algorithm [8]. *PAA* requires the number of segments for the time series as an input value. Moreover, all the partitions have an equal length. In our proposed approach each segment is calculated through the same idea as in *PAA* by using mean values. However, partitions will not necessarily have equal lengths. This difference can be stated as follows: let C be a time series with length n represented as a vector $\overline{C} = \overline{c_1}, ..., \overline{c_n}$ and $T = t_1, t_2, ..., t_m$ be the discretization scheme over word size, where $\{(t_i, t_{i+1}]\}$ is the time interval from segment i of $\overline{C}$, where the element i from $\overline{C}$ is given by: $\overline{c_i} = \frac{1}{(t_{i+1}-t_i)} \sum_{j=t_i+1}^{t_{i+1}} C_{tj}$

2.2 Evolutionary Programming

EP is a simple but powerful evolutionary algorithm where evolution is simulated at species level, i.e., no crossover is considered [5]. Instead, asexual reproduction is implemented by a mutation operator. The main steps in EP are:

1. Population initialization.
2. Evaluation of solutions.
3. Offspring generation by mutation.
4. Replacement.

From the steps mentioned above, the following elements must be defined so as to adapt EP to the time series discretization problem: (a) solution encoding, (b) fitness function to evaluate solutions, (c) mutation operator and (d) replacement mechanism. They are described below.

Solution Encoding. As in other evolutionary algorithms, in EP a complete solution of the problem must be encoded in each individual. A complete discretization scheme is encoded as shown in Fig. 2a, where the word size is encoded first with integer values, followed by the alphabet represented by real numbers, which must be sorted so as to apply the scheme to the time series data set [17] as shown in Fig. 2b.

Fitness Function. Different measures have been reported in the specialized literature to determine the quality of discretization schemes, such as information criterion [3], persistence state [14], information entropy maximization *(IEM)*, information gain, entropy maximization, Petterson-Niblett and minimum description length *(MDL)* [11,4]. Our fitness function, which aims to bias EP to promising regions of the search space, is based on three elements:

1. Classification accuracy (accuracy) based on entropy.
2. Strings reduction level (num_strings).
3. Compression level (num_cutpoints).

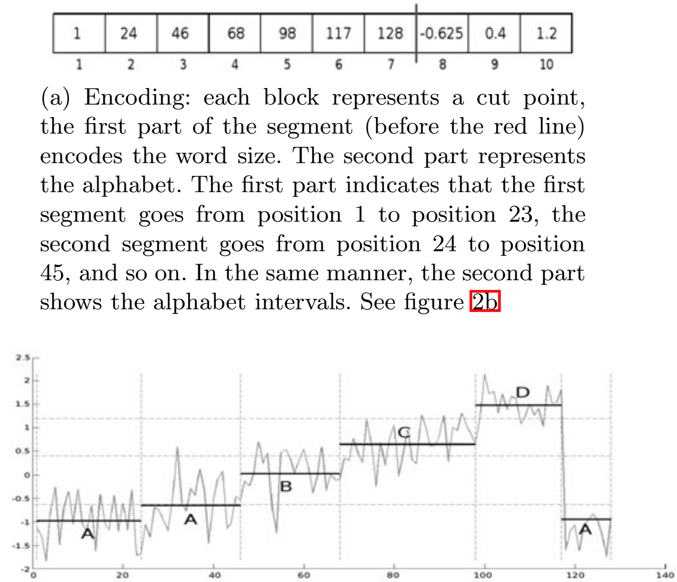
Those three values are normalized and added into one single value using the relationship in Eq. 1 for individual j in the population Pop.

$$Fitness(Pop_j) = (\alpha\ accuracy) + (\beta\ num\_strings) + (\gamma\ num\_cutpoints) \quad (1)$$

where: α, β y γ are the weights whose values determine the importance of each element.

The whole evaluation process for a given individual, i.e., discretization scheme, requires the following steps: First, the discretization scheme is applied over the complete time series data set S. Then, NS strings are obtained, where NS is equal to the number of time series from the data set S. A $m \times ns$ matrix called M is generated, where m is the number of different classes and ns is the number of different strings obtained. From this discretized data set, S_U is computed as the list of unique strings. Each of these strings has its own class label C. The first element of Eq. 1 (accuracy) is computed through entropy calculation over the columns of the matrix M as indicated in Equation 2.

$$accuracy = \sum_{j=0}^{\#S_U} Entropy(Col_j) \quad (2)$$



1	24	46	68	98	117	128	-0.625	0.4	1.2
1	2	3	4	5	6	7	8	9	10

(a) Encoding: each block represents a cut point, the first part of the segment (before the red line) encodes the word size. The second part represents the alphabet. The first part indicates that the first segment goes from position 1 to position 23, the second segment goes from position 24 to position 45, and so on. In the same manner, the second part shows the alphabet intervals. See figure 2b

(b) Decoding: the decoded solution from Fig. 2a, after sorting the values for the word size and the alphabet, is applied to a time series. The solution can be seen as a grid (i.e., the word size over the x-axis and the alphabet over the y-axis)

Fig. 2. Solution encoding-decoding

where: $\#S_U$ is the number of different strings and Col_j is the column j of the matrix M. The second element $num\_strings$ is calculated in Eq. 3

$$num\_strings = (\#S_U - \#C)/(N + \#C) \tag{3}$$

where: $\#S_U$ is the number of different strings, N is the number of time series in data set and $\#C$ is the number of existing classes. Finally, the third element $num\_cutpoints$ is computed in Eq. 4

$$num\_cutpoints = (size\_individual/(2 * length\_series)) \tag{4}$$

where: $size\_individual$ is the number of partitions (word size) that the particular discretization scheme has and $length\_series$ is the size of the original time series. In summary, the first element represents how well a particular individual (discretization scheme) is able to correctly classify the data base, the second element asses the complexity of the representation in terms of different patterns needed to encode the data, and the third element is a measure of the compression rate reached using a particular discretization scheme.

Mutation Operator. The mutation operator is applied to every individual in the population in order to generate one offspring per individual. We need a value

$N\_MUT \in [1, 2, 3]$ to define how many changes will be made to an individual. Each time an individual is mutated the N_MUT value is calculated. Each change consists on choosing a position of the vector defined in Fig. 2a and generate a new valid value at random.

Replacement Mechanism. The replacement mechanism consists on sorting the current population and their offspring by their fitness values in and letting the first half to survive for the next generation while the second half is eliminated.

The pseudocode of our EP algorithm to tackle the times series discretization problem is presented in Algorithm 1, where a population of individuals, i.e., valid schemes is generated at random. After that, each individual m generates one offspring by the mutation implemented as one to three random changes in the encoding. The set of current individuals Pop and the set of $Offspring$ are merged into one set called $Pop^{'}$ which is sorted based on fitness and the first-half remains for the next generation. The process finishes when a number of $MAXGEN$ generations is computed and the best discretization scheme is then used to discretize the data base to be classified by the *K-nearest neighbors (KNN)* algorithm.

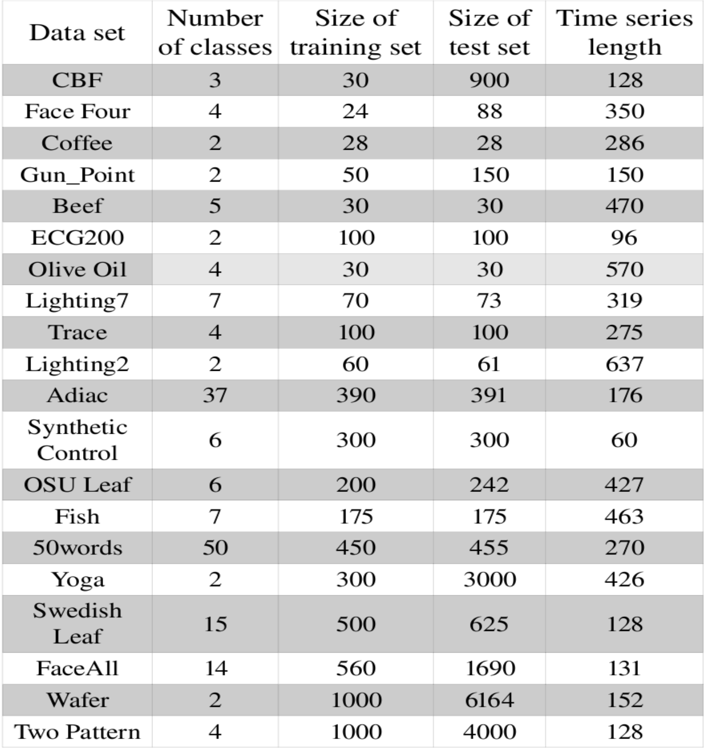
Algorithm 1 . EP pseudocode

```
Pop= $\emptyset$
for $m = 0$ to $popsize$ do
    Pop$_m$ = Valid_Scheme() %Generate individuals at random.
end for
for $k = 0$ to $MAXGEN$ do
    Offspring = $\emptyset$
    for $m = 0$ to $popsize$ do
        Offspring$_m$ = Mutation(Pop$_m$) %Create a new individual by mutation
    end for
    Pop$^{'}$ = Replacement(Pop + Offspring) %Select the best ones
    Pop=Pop$^{'}$
end for
```

3 Experiments and Results

The EP algorithm was tested on twenty data sets of the largest collection of time series data sets in the world, the UCR Time Series Classification/Clustering repository [10]. A summary of the features of each data set is presented in Table 1. The EP algorithm was executed with the following parameters experimentally found after preliminary tests: $popsize = 250$ and $MAXGEN = 50$, $\alpha = 0.9009$, $\beta = 0.0900$ and $\gamma = 0.0090$. When we ran the algorithm with other values for *popsize* and $MAXGEN$, we noticed that these ones worked better. If they were lower, the results would not find many possible solutions. And if they were higher, some solutions would be lost. About alpha parameter we saw that it could not have all the weight, even if it is the most important, because we would have many ties in some data sets. Practically, beta and gamma parameters avoid ties.

Table 1. Data sets used in the experiments



Data set	Number of classes	Size of training set	Size of test set	Time series length
CBF	3	30	900	128
Face Four	4	24	88	350
Coffee	2	28	28	286
Gun_Point	2	50	150	150
Beef	5	30	30	470
ECG200	2	100	100	96
Olive Oil	4	30	30	570
Lighting7	7	70	73	319
Trace	4	100	100	275
Lighting2	2	60	61	637
Adiac	37	390	391	176
Synthetic Control	6	300	300	60
OSU Leaf	6	200	242	427
Fish	7	175	175	463
50words	50	450	455	270
Yoga	2	300	3000	426
Swedish Leaf	15	500	625	128
FaceAll	14	560	1690	131
Wafer	2	1000	6164	152
Two Pattern	4	1000	4000	128

The quality of the solutions obtained by the EP algorithm was computed by using the best discretization scheme obtained for a set of five independent runs in the *k-nearest neighbors* classifier with $K = 1$. Other K values were tested ($K = 3$ and $K = 5$) but the performance decreased in all cases. Therefore, the results were not included in this paper. The low number of runs (5) is due to the time required (more than an hour) for the algorithm to process one single run for a given data set. The distance measure used in the *k-nearest neighbors* algorithm was Euclidean distance. The algorithms used for comparison were GENEBLA and SAX. The raw data was also included as a reference. Based on the fact that SAX requires the word length and alphabet as inputs, it was run using the parameters obtained by the EP algorithm and also by those obtained by GENEBLA.

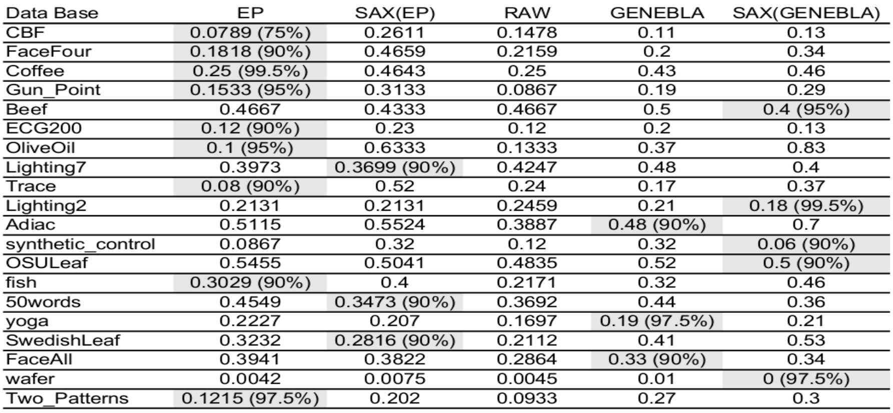
Table 2 summarizes the error rate on the twenty time series data sets for $K = 1$ for each evaluated algorithm (EP, SAX(EP), GENEBLA, SAX(GENEBLA)) and raw data. Values go from zero to one, where the lower means the better value. The values between parentheses indicate the confidence on the significance of the differences observed based on statistical tests applied to the samples of results per algorithm. In all cases the differences were significant.

From the results in Table 2 it can be noticed that different performances were provided by the compared algorithms. EP obtained the lowest error rate in nine data sets. On the other hand, GENEBLA had better results in just three data sets. Regarding the combination of EP and GENEBLA with SAX, slightly better results were observed with GENEBLA-SAX with respect to EP-SAX, where better results were obtained in five and three data sets, respectively.

It is worth noticing that EP provided its best performance in data sets with a lower number of classes, (between 2 and 4): CBF, Face Four, Coffee, Gun_Point, FGC200 and Two Pattern (Fish is the only exception). On the other hand, GENEBLA performed better in data sets with more classes (Adiac with 37 and Face All with 14). Another related interesting finding is that SAX seems to help EP to solve with the best performance of the compared approaches those data sets with a higher number of classes (Lighting7 with 7, 50words with 50 and Swedish Leaf with 15). In contrast, the combination of GENEBLA with SAX help the former to deal with some data sets with a lower number of classes (Beef with 5, Lighting2 with 2, Synthetic Control and OSU Leaf with 6 and Wafer with 2).

Finally, there was not a clear pattern about the algorithm with the best performance by considering the sizes of the training and test sets as well as the time series length.

Table 2. Error rate obtained by each compared approach in the 20 data sets. The best result for each data set is remarked with a gray background. Raw data is presented only as a reference.



Data Base	EP	SAX(EP)	RAW	GENEBLA	SAX(GENEBLA)
CBF	0.0789 (75%)	0.2611	0.1478	0.11	0.13
FaceFour	0.1818 (90%)	0.4659	0.2159	0.2	0.34
Coffee	0.25 (99.5%)	0.4643	0.25	0.43	0.46
Gun_Point	0.1533 (95%)	0.3133	0.0867	0.19	0.29
Beef	0.4667	0.4333	0.4667	0.5	0.4 (95%)
ECG200	0.12 (90%)	0.23	0.12	0.2	0.13
OliveOil	0.1 (95%)	0.6333	0.1333	0.37	0.83
Lighting7	0.3973	0.3699 (90%)	0.4247	0.48	0.4
Trace	0.08 (90%)	0.52	0.24	0.17	0.37
Lighting2	0.2131	0.2131	0.2459	0.21	0.18 (99.5%)
Adiac	0.5115	0.5524	0.3887	0.48 (90%)	0.7
synthetic_control	0.0867	0.32	0.12	0.32	0.06 (90%)
OSULeaf	0.5455	0.5041	0.4835	0.52	0.5 (90%)
fish	0.3029 (90%)	0.4	0.2171	0.32	0.46
50words	0.4549	0.3473 (90%)	0.3692	0.44	0.36
yoga	0.2227	0.207	0.1697	0.19 (97.5%)	0.21
SwedishLeaf	0.3232	0.2816 (90%)	0.2112	0.41	0.53
FaceAll	0.3941	0.3822	0.2864	0.33 (90%)	0.34
wafer	0.0042	0.0075	0.0045	0.01	0 (97.5%)
Two_Patterns	0.1215 (97.5%)	0.202	0.0933	0.27	0.3

4 Conclusions and Future Work

We presented a novel time series discretization algorithm based on EP. The proposed algorithm was able to automatically find the parameters for a good discretization scheme considering the optimization of accuracy and compression rate. Moreover, and as far as we know, this is the first approach that considers the world length and the alphabet optimization at the same time. A simple mutation operator was able to sample the search space by generating new and competitive solutions. Our EP algorithm is easy to implement and the results obtained in 20 different data sets were highly competitive with respect to previously proposed methods including the raw data, i.e., the original time series, which means that the EP algorithm is able to obtain the important information of a continuous time series and disregards unimportant data. The EP algorithm provided a high performance with respect to GENEBLA in problems with a low number of classes. However, if EP is combined with SAX, the approach is able to outperform GENEBLA and also GENEBLA-SAX in problems with a higher number of classes.

The future work consists on a further analysis of the EP algorithm such as the effect of the weights in the search as well as the number of changes in the mutation operator. Furthermore, other classification techniques (besides *k-nearest neighbors*) and other evolutive approaches like PSO need to be tested. Finally, Pareto dominance will be explored with the aim to deal with the three objectives considered in the fitness function [15].

References

1. García-López, D.-A., Acosta-Mesa, H.-G.: Discretization of Time Series Dataset with a Genetic Search. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS, vol. 5845, pp. 201–212. Springer, Heidelberg (2009)
2. Acosta-Mesa, H.G., Nicandro, C.R., Daniel-Alejandro, G.-L.: Entropy Based Linear Approximation Algorithm for Time Series Discretization. In: Advances in Artificial Intelligence and Applications, vol. 32, pp. 214-224. Research in Computers Science
3. Dimitrova, E.S., McGee, J., Laubenbacher, E.: Discretization of Time Series Data, (2005) eprint arXiv:q-bio/0505028.
4. Fayyad, U., Irani, K.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence (1993)
5. Fogel, L.: Intelligence Through Simulated Evolution. Forty years of Evolutionary Programming (Wiley Series on Intelligent Systems) (1999)
6. García-López D.A.: Algoritmo de Discretizaciǿn de Series de Tiempo Basado en Entropía y su Aplicaciǿn en Datos Colposcǿpicos. Universidad Veracruzana (2007)
7. Han, J., Kamber, M.: Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems) (2001)
8. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. ACM Trans. Database Syst. (2002)

9. Keogh, E., Lonardi, S., Ratanamabatana, C.A.: Towards parameter-free data mining. In: Proceedings of Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2001)
10. Keogh, E., Xi, C., Wei, L., Ratanamabatana, C.A.: The UCR Time Series Classification/Clustering Homepage (2006), http://www.cs.ucr.edu/~eamonn/time_series_data/
11. Kurgan, L., Cios, K.: CAIM Discretization Algorithm. IEEE Transactions On Knowledge And Data Engineering (2004)
12. Last, M., Kandel, A., Bunke, H.: Data mining in time series databases. World Scientific Pub. Co. Inc., Singapore (2004)
13. Lin, J., Keogh, E., Lonardi, S., Chin, B.: A symbolic representation of time series, with implications for streaming Algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (2003)
14. Mörchen, F., Ultsch, A.: Optimizing Time Series Discretization for Knowledge Discovery. In: Proceeding of the Eleventh ACM SIGKDD international Conference on Knowledge Discovery in Data Mining (2005)
15. Kalyanmoy, D., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation (2002)
16. Trevor, H., Tibshirani, R., Friedman, J.: The elements of Statistical Learning. Springer, Heidelberg (2009)
17. Chiu, C., Nanh, S.C.: An adapted covering algorithm approach for modeling airplanes landing gravities. Expert Systems with Applications 26, 443–450 (2004)

Clustering of Heterogeneously Typed Data with Soft Computing – A Case Study

Angel Kuri-Morales[1], Daniel Trejo-Baños[2], and Luis Enrique Cortes-Berrueco[2]

[1] Instituto Tecnológico Autónomo de México, Río Hondo No. 1 México D.F. México
akuri@itam.mx
[2] Universidad Nacional Autónoma de México, Apartado Postal 70-600,
Ciudad Universitaria,México D.F., México
{l.cortes,d.trejo}@uxmcc2.iimas.unam.mx

Abstract. The problem of finding clusters in arbitrary sets of data has been attempted using different approaches. In most cases, the use of metrics in order to determine the adequateness of the said clusters is assumed. That is, the criteria yielding a measure of quality of the clusters depends on the distance between the elements of each cluster. Typically, one considers a cluster to be adequately characterized if the elements within a cluster are close to one another while, simultaneously, they appear to be far from those of different clusters. This intuitive approach fails if the variables of the elements of a cluster are not amenable to distance measurements, i.e., if the vectors of such elements cannot be quantified. This case arises frequently in real world applications where several variables (if not most of them) correspond to categories. The usual tendency is to assign arbitrary numbers to every category: to encode the categories. This, however, may result in spurious patterns: relationships between the variables which are not really there at the offset. It is evident that there is no truly valid assignment which may ensure a universally valid numerical value to this kind of variables. But there is a strategy which guarantees that the encoding will, in general, not bias the results. In this paper we explore such strategy. We discuss the theoretical foundations of our approach and prove that this is the best strategy in terms of the statistical behavior of the sampled data. We also show that, when applied to a complex real world problem, it allows us to generalize soft computing methods to find the number and characteristics of a set of clusters. We contrast the characteristics of the clusters gotten from the automated method with those of the experts.

Keywords: Clustering, Categorical variables, Soft computing, Data mining.

1 Introduction

1.1 Clustering

Clustering can be considered the most important unsupervised learning problem. As every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. In this particular case it is of relevance because we attempt to characterize sets of arbitrary data trying not to start from preconceived measures of what

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 235–248, 2011.

makes a set of characteristics relevant. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

When the similarity criterion is distance two or more objects belong to the same cluster if they are “close” according to a given distance. This is called distance-based clustering. Another kind of clustering is conceptual clustering where two or more objects belong to the same cluster if this one defines a concept common to all those objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures [1,2,7,9]. Our contention is that conceptual clustering leads to biased criteria which have lead to the unsuccessful generalization properties of the models proposed in the past.

1.2 The Need to Encode

In recent years there has been an increasing interest to analyze categorical data in a data warehouse context where data sets are rather large and may have a high number of categorical dimensions [4,6,8,15]. However, many traditional techniques associated to the exploration of data sets assume the attributes have continuous data (covariance, density functions, PCA, etc.). In order to use these techniques, the categorical attributes have to be discarded, although they are potentially loaded with valuable information. With our technique the categorical attributes are encoded into numeric values in such a way that spurious correlations are avoided and the data can be handled as if it were numeric.

In [5] the authors propose a framework designed for categorical data analysis that allows the exploration of this kind of data with techniques that are only applicable to continuous data sets. By means of what the authors call “separability statistics", e.g. matching values with instances in a reference data set, they map any collection of categorical instances to a multidimensional continuous space. This way, instances similar to a reference data set, that could be the original dataset itself, will occupy the same region occupied by instances from the reference dataset and instances that are different will tend to occupy other regions. This mapping enables visualizing the categorical data using techniques that are applicable to continuous data. Their framework can be used in the context of several data mining tasks such as outlier detection, clustering and classification. In [3], the authors show how the choice of a similarity measure affects performance. By contrast, our encoding technique maps the categorical data to a numerical domain. The mapping is done avoiding the transmission of spurious correlations to the corresponding encoded numerical data. Once the data is numerically encoded, techniques applicable to continuous data can be used.

Following a different approach, in [11] the authors propose a distance named “distance hierarchy", based on concept hierarchies [10] extended with weights, in order to measure the distance between categorical values. This type of measure allows the use of data mining techniques based on distances, e.g. clustering techniques, when dealing with mixed data, numerical and categorical. With our technique, by encoding categorical data into numeric values, we can use then the traditional distance computations avoiding the need to figure out different ways to compute distances. Another approach is followed in [13]. The authors propose a measure in order to quantify dissimilarity of objects by using distribution information of data correlated to each categorical value. They propose a method to uncover intrinsic relationship of values by using a dissimilarity measure referred to as Domain Value Dissimilarity (DVD). This measure is independent of any specific algorithm so that it can be applied to clustering algorithms that require a distance measure for objects. In [14] the authors present a process for quantification (i.e. quantifying the categorical variables - assigning order and distance to the categories) of categorical variables in mixed data sets, using Multiple Correspondence Analysis, a technique which may be seen as the counterpart of principal component analysis for categorical data. An interactive environment is provided, in which the user is able to control and influence the quantification process and analyze the result using parallel coordinates as a visual interface. For other possible clustering methods the reader is referred to [12,16,17,18,24].

2 Unbiased Encoding of Cathegorical Variables

We now introduce an alternative which allows the generalization of numerical algorithms to encompass categorical variables. Our concern is that such encoding:

a) Does not induce spurious patterns
b) Preserves legal patters, i.e. those present in the original data.

By "spurious" patterns we mean those which may arise by the artificial distance induced by our encoding. On the other hand, we do not wish to filter out those patterns which are present in the categories. If there is an association pattern in the original data, we want to preserve this association and, furthermore, we wish to preserve it in the same way as it presents itself in the original data. The basic idea is simple: "Find the encoding which best preserves a measure of similarity between all numerical and categorical variables".

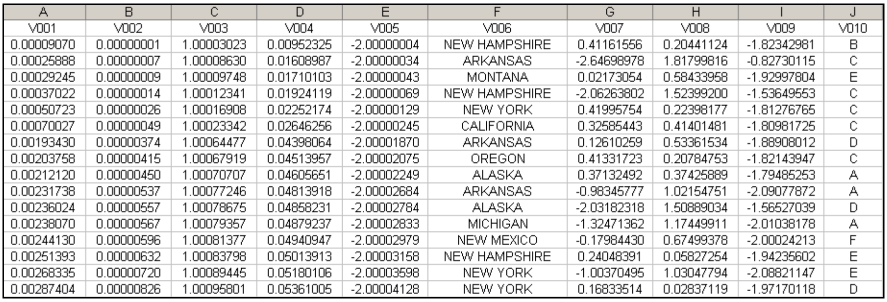
In order to do this we start by selecting Pearson's correlation as a measure of linear dependence between two variables. Higher order dependencies will be hopefully found by the clustering algorithms. This is one of several possible alternatives. The interested reader may see [25,26]. Its advantage is that it offers a simple way to detect simple linear relations between two variables. Its calculation yields "r", Pearson's correlation, as follows:

$$r = \frac{N\sum XY - \sum X \sum Y}{\sqrt{\left[N\sum X^2 - \left(\sum X\right)^2\right]\left[N\sum Y^2 - \left(\sum Y\right)^2\right]}} \tag{1}$$

Where variables X and Y are analyzed to search their correlation, i.e. the way in which one of the variables changes (linearly) with relation to the other. The values of "r" in (1) satisfy $-1 \leq r \leq +1$. What we shall do is search for a code for categorical variable A such that the correlation calculated from such encoding does not yield a significant difference with any of the possible encodings of all other categorical or numerical variables.

2.1 Exploring the Correlations

To exemplify let us assume that our data consists of only 10 variables. In this case there are 5,000 objects (or 10-dimensional vectors) in the data base. A partial view is shown in figure 1. Notice that two of the variables (V006 and V010) are categorical, whereas the rest are numerical.



A	B	C	D	E	F	G	H	I	J
V001	V002	V003	V004	V005	V006	V007	V008	V009	V010
0.00009070	0.00000001	1.00003023	0.00952325	-2.00000004	NEW HAMPSHIRE	0.41161556	0.20441124	-1.82342981	B
0.00025888	0.00000007	1.00008630	0.01608987	-2.00000034	ARKANSAS	-2.64698978	1.81799816	-0.82730115	C
0.00029245	0.00000009	1.00009748	0.01710103	-2.00000043	MONTANA	0.02173054	0.58433958	-1.92997804	E
0.00037022	0.00000014	1.00012341	0.01924119	-2.00000069	NEW HAMPSHIRE	-2.06263802	1.52399200	-1.53649553	C
0.00050723	0.00000026	1.00016908	0.02252174	-2.00000129	NEW YORK	0.41995754	0.22398177	-1.81276765	C
0.00070027	0.00000049	1.00023342	0.02646256	-2.00000245	CALIFORNIA	0.32585443	0.41401481	-1.80981725	C
0.00193430	0.00000374	1.00064477	0.04398064	-2.00001870	ARKANSAS	0.12610259	0.53361534	-1.88908012	D
0.00203758	0.00000415	1.00067919	0.04513957	-2.00002075	OREGON	0.41331723	0.20784753	-1.82143947	C
0.00212120	0.00000450	1.00070707	0.04605651	-2.00002249	ALASKA	0.37132492	0.37425889	-1.79485253	A
0.00231738	0.00000537	1.00077246	0.04813918	-2.00002684	ARKANSAS	-0.98345777	1.02154751	-2.09077872	A
0.00236024	0.00000557	1.00078675	0.04858231	-2.00002784	ALASKA	-2.03182318	1.50889034	-1.56527039	D
0.00238070	0.00000567	1.00079357	0.04879237	-2.00002833	MICHIGAN	-1.32471362	1.17449911	-2.01038178	A
0.00244130	0.00000596	1.00081377	0.04940947	-2.00002979	NEW MEXICO	-0.17984430	0.67499378	-2.00024213	F
0.00251393	0.00000632	1.00083798	0.05013913	-2.00003158	NEW HAMPSHIRE	0.24048391	0.05827254	-1.94235602	E
0.00268335	0.00000720	1.00089445	0.05180106	-2.00003598	NEW YORK	-1.00370495	1.03047794	-2.08821147	E
0.00287404	0.00000826	1.00095801	0.05361005	-2.00004128	NEW YORK	0.16833514	0.02837119	-1.97170118	D

Fig. 1. Mixed type data

We define the *i-th instance* of a categorical variable V_X as one possible value of variable X. For example, if variable V006 takes 28 different names, one instance is "NEW HAMPSHIRE", another instance is "ARKANSAS" and so on. We denote the number of variables in the data as V. Further, we denote with r_{ik} Pearson's correlation between variables i and k. We would like to a) Find the mean μ of the correlation's probability distribution for all categorical variables by analyzing all possible combinations of codes assignable to the categorical variables (in this example V006 and V010) plus the original (numerical) values of all non-categorical variables. b) Select the codes for the categorical variables which yield the closest value to μ. The rationale is that the absolute typical value of μ is the one devoid of spurious patterns *and* the one preserving the legal patterns. In the algorithm to be discussed next the following notation applies:

N ← number of elements in the data
V ← number of categorical variables
V[i] ← the i-th variable
Ni ← number of instances of V[i]
$\overline{r_j}$ ← the mean of the j-th sample
S ← sample size of a mean
$\mu_{\overline{r}}$ ← mean of the correlation's distribution of means
$\sigma_{\overline{r}}$ ← standard deviation of the correlation's distribution of means

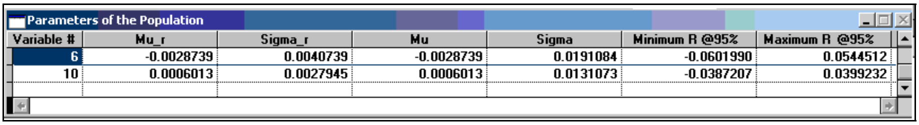
Algorithm A1.
Optimal Code Assignment for Categorical Variables

```
01  for i=1 to V
02    j ← 0
03    do while $\overline{r_j}$ is not distributed normally
04      for k=1 to S
05        Assign a code for variable V[i]
06        Store this code
07        $\ell$ ← integer random number (1≤$\ell$≤ V; $\ell$≠i)
08        if variable V[$\ell$] is categorical
09          Assign a code for variable V[$\ell$]
10        endif
11        $r_k = \frac{N\sum XY - \sum X \sum Y}{\sqrt{\left[N\sum X^2 - \left(\sum X\right)^2\right]\left[N\sum Y^2 - \left(\sum Y\right)^2\right]}}$
12      endfor
13      Calculate $\overline{r_j}$ ← $= \frac{1}{S}\sum_{k=1}^{S} r_k$
14      j ← j+1
15    enddo
16    $\mu = \mu_{\overline{r}}$; the mean of the correlation's distribution
17    $\sigma = \sqrt{SS}\cdot\sigma_{\overline{r}}$; the std. dev. of the correlation's distribution
18    Select the code for V[i] which yields the $r_k$ closest to $\mu$
19  endfor
```

For simplicity, in the formula of line (11), X stands for variable V[i] and Y stands for variable V[ℓ]. Of course it is impossible to consider all codes, let alone all possible combinations of such codes. Therefore, in algorithm A1 we set a more modest goal and adopt the convention that to *Assign a Code* [as in lines (05) and (09)] means that we restrict ourselves to the combinations of integers between 1 and Ni (recall that *Ni* is the number different values of variable *i* in the data). Still, there are Ni! possible ways to assign a code to categorical variable *i* and Ni! x Nj! possible encodings of two categorical variables *i* and *j*. An exhaustive search is, in general, out of the question. Instead, we take advantage of the fact that, regardless of the way a random variable distributes (here the value of the random encoding of variables *i* and *j* results in correlation r_{ij} which is a random variable itself) the *means* of sufficiently large samples very closely approach a normal distribution. Furthermore, the mean value of a sample of means $\mu_{\overline{r}}$ and its standard deviation $\sigma_{\overline{r}}$ are related to the mean μ and standard deviation σ of the original distribution by $\mu = \mu_{\overline{r}}$ and $\sigma = \sqrt{SS}\cdot\sigma_{\overline{r}}$. What a *sufficiently large* sample means is a matter of convention and here we made S=25 which is a reasonable choice. Therefore, the loop between lines (03) and (15) is guaranteed to end. In our implementation we split the area under the normal curve in deciles and then used a χ^2 goodness-of-fit test with p=0.05 to determine that normality has been achieved. This approach is directed to avoid arbitrary assumptions regarding the correlation's distribution and, therefore, not selecting a sample size to establish the reliability of our results. Rather, the algorithm determines at what point the proper value of μ has been reached. Furthermore, from Chebyshev's theorem, we know that

$$P(\mu - k\sigma \leq X \leq \mu + k\sigma) \geq 1 - \frac{1}{k^2} \tag{2}$$

If we make k=3 and assume a symmetrical distribution, the probability of being within three σ's of the mean is roughly 0.95. We ran our algorithm for the data of the example and show in figure 5 the values that were obtained.



Variable #	Mu_r	Sigma_r	Mu	Sigma	Minimum R @95%	Maximum R @95%
6	-0.0028739	0.0040739	-0.0028739	0.0191084	-0.0601990	0.0544512
10	0.0006013	0.0027945	0.0006013	0.0131073	-0.0387207	0.0399232

Fig. 2. Values of categorical encoding for variables 6 and 10

In the program corresponding to figure 2, Mu_r and Sigma_r denote the mean and standard deviation of the distribution of means; Mu and Sigma denote the corresponding parameters for the distribution of the correlations and the titles "Minimum R @95%" and "Maximum R@95%" denote the smallest and largest values at ±3 σ's from the mean. In this case, the typical correlation is close to zero, denoting no first order patterns in the data. With probability 0.95 the typical correlation for variable 6 lies in an interval of size 0.1147 while the corresponding value for variable 10 lies in an interval of size 0.0786. Three other issues remain to be clarified.

1) To *Assign a code* to V[i] means that we generate a sequence of numbers between 1 and Ni and then randomly assign a one of these numbers to every different instance of V[i].

2) To *Store the code* [as in line (06)] means NOT that we store the assigned code (for this would imply storing a large set of sequences). Rather, we store the value of the calculated correlation along with the root of the pseudo random number generator from which the assignment was derived.

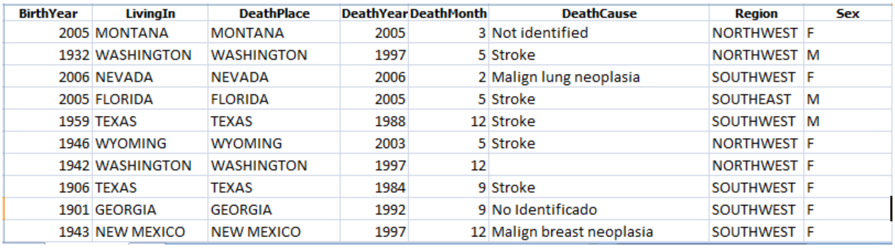
3) Thereafter, selecting the best code (i.e. the one yielding a correlation whose value is closest to μ) as in line (18) is a simple matter of recovering the root of the pseudo random number generator and regenerating the original random sequence from it.

3 Case Study: Profile of the Causes of Death of a Population

In order to illustrate our method we analyzed a data base corresponding to the life span and cause of death of 50,000 individuals between the years of 1900 and 2007. The confidentiality of the data has been preserved by changing the locations and regions involved. Otherwise data are a faithful replica of the original.

3.1 The Data Base

This experiment allowed us to compare the interpretation of the human experts with the one resulting from our analysis. The database contains 50,000 tuples consisting of 11 fields: BirthYear, LivingIn, DeathPlace, DeathYear, DeathMonth, DeathCause, Region, Sex, AgeGroup, AilmentGroup and InterestGroup. A very brief view of 8 of the 11 variables is shown in figure 3.



BirthYear	LivingIn	DeathPlace	DeathYear	DeathMonth	DeathCause	Region	Sex
2005	MONTANA	MONTANA	2005	3	Not identified	NORTHWEST	F
1932	WASHINGTON	WASHINGTON	1997	5	Stroke	NORTHWEST	M
2006	NEVADA	NEVADA	2006	2	Malign lung neoplasia	SOUTHWEST	F
2005	FLORIDA	FLORIDA	2005	5	Stroke	SOUTHEAST	M
1959	TEXAS	TEXAS	1988	12	Stroke	SOUTHWEST	M
1946	WYOMING	WYOMING	2003	5	Stroke	NORTHWEST	F
1942	WASHINGTON	WASHINGTON	1997	12		NORTHWEST	F
1906	TEXAS	TEXAS	1984	9	Stroke	SOUTHWEST	F
1901	GEORGIA	GEORGIA	1992	9	No Identificado	SOUTHWEST	F
1943	NEW MEXICO	NEW MEXICO	1997	12	Malign breast neoplasia	SOUTHWEST	F

Fig. 3. Partial view of the data base

The last variable (InterestGroup) corresponds to interest groups identified by human healthcare experts in this particular case. This field corresponds to a heuristic clustering of the data and will be used for the final comparative analysis of resulting clusters. It will not be included either in the data processing nor the data mining activities. Therefore, our working data base has 10 dimensions.

The first thing to notice is that there are no numeric variables. BirthYear, DeathYear and DeathMonth are dates (clearly, they represent the date of birth, year and month of death respectively). "Region" represents the place where the death took place. DeathCause and AilmentGroup are the cause of death and the illness group to which the cause of death belongs.

3.2 Preprocessing the Information

In order to process the information contained in the data base we followed the next methodology:

- At the offset we applied algorithm A1 and, once the coding process was finished we got a set of 10 codes, each code with a number of symbols corresponding to the cardinality of the domain of the variable.
- Each column of the data base is encoded.
- We get the correlation between every pair of variables. If the correlation between two columns is large only one of them is retained.
- We assume no prior knowledge of the number of the clusters and, therefore, resorted to the Fuzzy C-Means algorithm and the elbow criterion to determine it [see 19, 20]. For a sample of K objects divided in c classes (where μ_{ik} is the membership of an object k to class i) we determine the partition coefficient (pc) and the partition entropy (pe) from formulas (3) and (4) respectively [see 21, 22, 23].

$$pc = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{c}\mu_{ik} \tag{3}$$

$$pe = -\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{c}\mu_{ik}ln(\mu_{ik}) \tag{4}$$

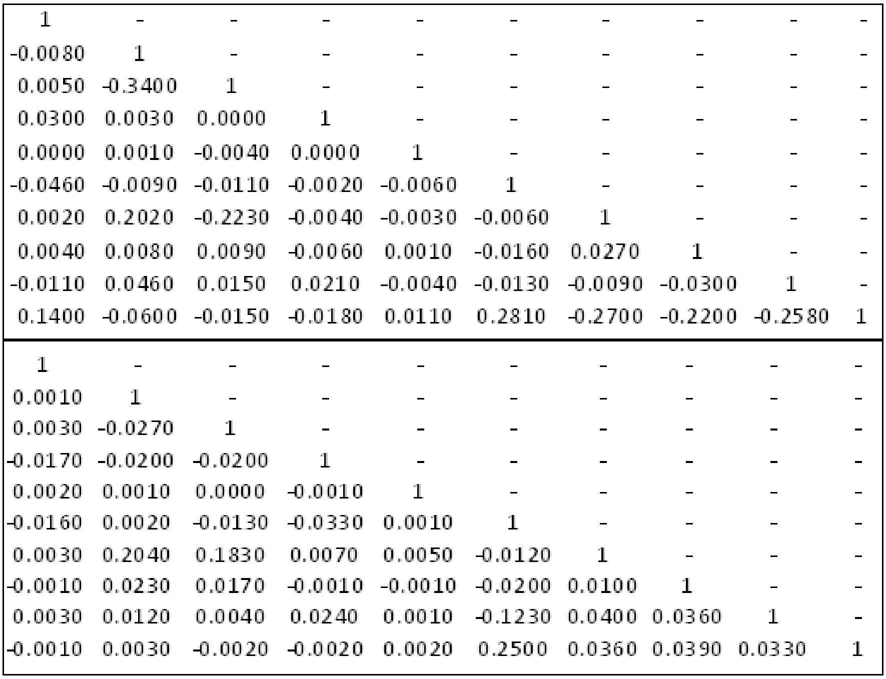
3.3 Processing the Information

We process the information with two unsupervised learning techniques: Fuzzy c-means and Kohonen's SOM.

There is only one difference in the pre-process phase. For the Kohonen's SOM case a filtering of the data set was conducted. It was found that in several tuples the death date precedes birth date resulting in an inconsistent representation of reality. The data set was scanned and all the cases presenting the error were deleted. As a result of this action the original set was reduced from 500,000 tuples to 485,289.

In both cases the categorical data was encoded to numbers, we obtained the correlation between the variables, Figure 4 presents the correlation matrix.

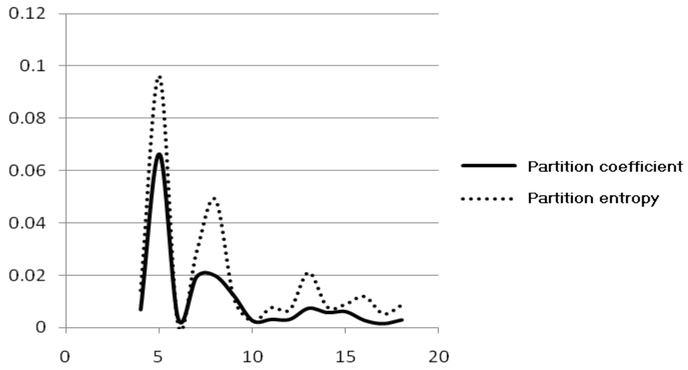
The largest absolute correlation does not exceed 0.3. Hence, there are no strongly correlated variables. It is important to notice that the highest correlations are consistent with reality: (1,6) Birth Place – Region of the country,(5,9) Pathology – Pathology Group.



1	-	-	-	-	-	-	-	-	-
-0.0080	1	-	-	-	-	-	-	-	-
0.0050	-0.3400	1	-	-	-	-	-	-	-
0.0300	0.0030	0.0000	1	-	-	-	-	-	-
0.0000	0.0010	-0.0040	0.0000	1	-	-	-	-	-
-0.0460	-0.0090	-0.0110	-0.0020	-0.0060	1	-	-	-	-
0.0020	0.2020	-0.2230	-0.0040	-0.0030	-0.0060	1	-	-	-
0.0040	0.0080	0.0090	-0.0060	0.0010	-0.0160	0.0270	1	-	-
-0.0110	0.0460	0.0150	0.0210	-0.0040	-0.0130	-0.0090	-0.0300	1	-
0.1400	-0.0600	-0.0150	-0.0180	0.0110	0.2810	-0.2700	-0.2200	-0.2580	1

1	-	-	-	-	-	-	-	-	-
0.0010	1	-	-	-	-	-	-	-	-
0.0030	-0.0270	1	-	-	-	-	-	-	-
-0.0170	-0.0200	-0.0200	1	-	-	-	-	-	-
0.0020	0.0010	0.0000	-0.0010	1	-	-	-	-	-
-0.0160	0.0020	-0.0130	-0.0330	0.0010	1	-	-	-	-
0.0030	0.2040	0.1830	0.0070	0.0050	-0.0120	1	-	-	-
-0.0010	0.0230	0.0170	-0.0010	-0.0010	-0.0200	0.0100	1	-	-
0.0030	0.0120	0.0040	0.0240	0.0010	-0.1230	0.0400	0.0360	1	-
-0.0010	0.0030	-0.0020	-0.0020	0.0020	0.2500	0.0360	0.0390	0.0330	1

Fig. 4. Correlation Matrix (up fuzzy z-means, down Kohonen's SOM)



To determine the number of clusters we applied the fuzzy c-means algorithm to our coded sample. We experimented with 17 different possibilities (assuming from 2 to 18 clusters) for the fuzzy c-means case and with 30 different possibilities (from 2 to 31 clusters) for the Kohonen's SOM case. In figure 5 it is noticeable that the largest change occurs between 4 and 5 clusters for the first case and between 3 and 4 for the second case. In order to facilitate the forthcoming process we selected 4 clusters (fuzzy c-means case) and for variety, we picked 3 clusters in the other case. This first approach may be refined as discussed in what follows.

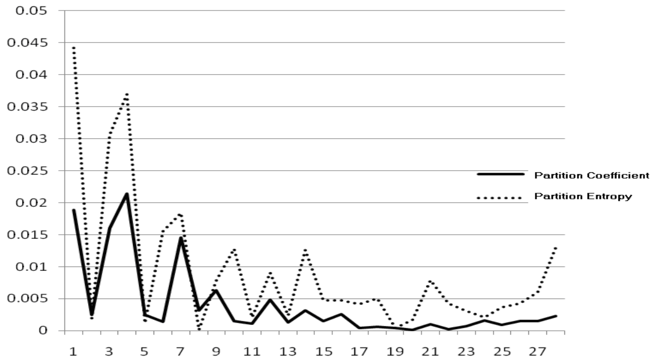


Fig. 5. Second differences graph (up fuzzy z-means, down Kohonen's SOM)

Fuzzy c-Means

Once the number of clusters is determined fuzzy c-means was applied to determine the cluster's centers. The result of the method, shown by the coordinates of the cluster centers is presented in figure 6. A brief graph showing the composition of the clusters centers can be seen in figure 10.

As can be seen in figure 7, the values for BirthYear and DeathCause are the ones that change the most within the cluster centers. An intuitive explanation is that the date of birth (and consequently the age) has had direct influence on the cause of death. The next step was a recall of the data. We grouped the tuples in one of the four classes, the one for which the tuple has the largest membership value. Now we achieve the classification of tuples on four crisp clusters. The clusters may then be analyzed individually.

C	Birth Year	Living In	Death Place	Death Year	Death Month	Death Cause	Region	Sex	Age-Group	Ailment Group
1	19.038	15.828	17.624	16.493	6.446	62.989	2.960	0.498	10.461	5.181
2	59.085	15.730	17.685	15.223	6.432	68.0876	2.970	0.507	10.464	5.611
3	58.874	15.980	17.355	15.576	6.427	28.632	2.959	0.465	10.671	3.860
4	106.692	15.646	17.613	17.211	6.453	64.647	3.026	0.492	10.566	5.317

Fig. 6. Clusters centers

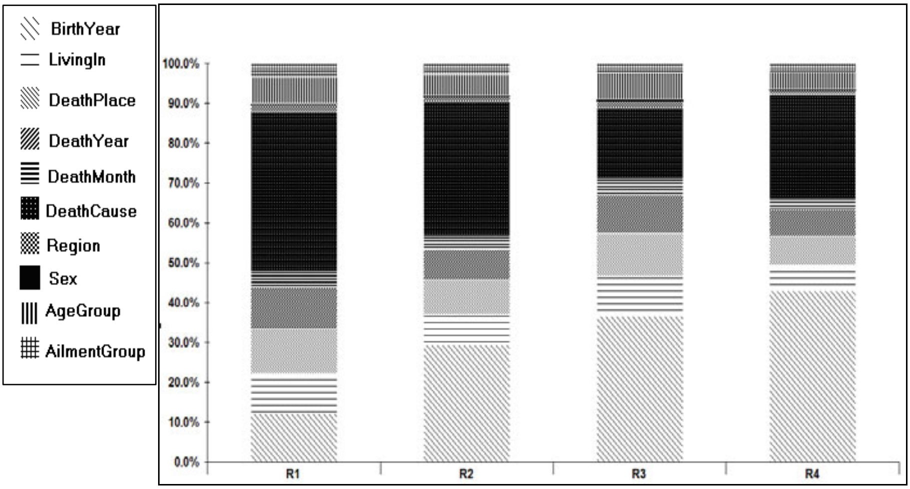


Fig. 7. Composition of the clusters

Limitations of space allow us to present, only, limited examples. In figure 8 we show the results for cluster 2. From this analysis various interesting facts were observed. The values of the means tend to be very close between all clusters in all variables except BirthYear and DeathCause. Cluster 2 has a mean for BirthYear close to that of cluster 3, but the mean of DeathCause is very different. Some very brief and simple observations follow.

Cluster 2	Birth Year	Living In	Death Place	Death Year	Death Month	Death Cause	Region	Sex	Age Group	Ailment Group
Mean	58.91	15.80	17.70	15.49	6.42	72.77	3.02	0.51	10.47	5.46
Mode	52	25	20	4	7	68	3	1	11	1
Variance	146.53	97.48	73.77	112.02	13.54	201.69	2.68	0.25	5.92	16.73
S.Deviation	12.10	9.87	8.59	10.58	3.68	14.20	1.64	0.50	2.43	4.09
Range	52.00	31.00	32.00	34.00	12.00	67.00	5.00	1.00	14.00	12.00
Skewness	0.83	0.49	-1.68	1.77	-1.19	4.83	-2.98	-0.31	-10.48	1.39
Kurtosis	1.44E+06	1.03E+06	1.46E+06	1.25E+06	1.30E+06	1.95E+06	1.53E+06	659795.45	4.75E+06	1.14E+06

Fig. 8. Basic Statistics of cluster number two

In cluster 1, for instance, the mode of BirthYear is 4, whose decoded value is the year 2006. The mode for DeathYear is 15 (decoded value 2008) and DeathCause corresponds to missing data. In cluster 2 the mode for BirthYear is 52, (1999), the mode for DeathCause is 68 (Diabetes type2). In cluster 3 the mode for BirthYear is 58 (2007 when decoded). For DeathCause the mode is 28 which correspondes to heart stroke. In cluster 4, the value of the mode for BirthYear is 4 (which corresponds to the year of 1900).

Kohonen's SOM

For this case we attempted to interpret the results according to the values of the mean, we rounded the said values for BirthYear and DeathCause and obtained the following decoded values:

- For cluster 1 the decoded values of the mean for BirthYear and DeathCause correspond to "1960" and cancer.
- In cluster 2 the values are "1919" and Pneumonia
- In cluster 3 the values are "1923" and Heart stroke

Interestingly, this approach seems to return more meaningful results than the mode based approach, by noting that people in different age groups die of different causes.

SOMs results were, as expected, similar the ones gotten from fuzzy C-means. However, when working with SOMs it is possible to split the clusters into subdivisions by increasing the number of neurons.

3.4 Clusters Proposed by Human Experts

Finally we present the general statistics for the clusters proposed by human experts as defined in the last column of the database. In the experts' opinion, there are only three clusters (see figure 9).

Cluster 1	**Birth Year**	**Living In**	**Death Place**	**Death Year**	**Death Month**	**Death Cause**	**Region**	**Sex**	**Age Group**	**Ailment Group**
Mean	33.82	15.38	16.96	15.84	5.46	37.83	1.88	0.51	10.23	7.78
Variance	313.98	54.54	81.22	78.85	11.44	423.76	1.85	0.25	13.99	43.26
Mode	4	25	20	15	7	28	3	0	11	1
S.Deviation	37.23	9.87	8.71	9.97	3.68	26.59	1.60	0.50	2.35	4.73
Range	129.00	31.00	32.00	34.00	12.00	117.00	5.00	1.00	14.00	12.00
Skewness	4.25	4.39	-10.89	5.45	-9.23	3.19	-22.26	6.35	-94.55	28.92
Kurtosis	32E+06	26E+06	36E+06	32E+06	33E+06	34E+06	40E+06	17E+06	14E+06	28E+06
Cluster 2	**Birth Year**	**Living In**	**Death Place**	**Death Year**	**Death Month**	**Death Cause**	**Region**	**Sex**	**Age Group**	**Ailment Group**
Mean	59.28	16.27	17.72	16.35	6.38	71.69	2.93	0.62	10.60	6.61
Mode	4	25	20	15	7	69	3	1	11	7
Variance	1233.71	98.22	71.74	106.25	13.60	103.20	2.84	0.24	5.32	1.34
S.Deviation	35.12	9.91	8.47	10.31	3.69	10.16	1.68	0.49	2.31	1.16
Range	129.00	31.00	32.00	34.00	12.00	114.00	5.00	1.00	14.00	12.00
Skewness	0.59	0.04	-1.66	0.82	-1.09	13.88	-2.59	-2.42	-9.23	-12.16
Kurtosis	7.6E+06	5.5E+06	8.2E+06	6.8E+06	7.0E+06	3.7E+07	7.8E+06	4.4E+06	2.6E+07	3.4E+07
Cluster 3	**Birth Year**	**Living In**	**Death Place**	**Death Year**	**Death Month**	**Death Cause**	**Region**	**Sex**	**Age Group**	**Ailment Group**
Mean	63.08	16.59	17.95	16.83	6.48	72.58	2.71	0.54	9.76	10.97
Mode	52	25	20	18	7	69	0	1	11	11
Variance	1340.82	95.95	65.18	111.74	13.25	76.32	3.23	0.25	10.33	0.24
S.Deviation	36.62	9.80	8.07	10.57	3.64	8.74	1.80	0.50	3.21	0.49
Range	129.00	31.00	32.00	34.00	12.00	95.00	5.00	1.00	14.00	11.00
Skewness	5.72 E-02	-8.21 E-02	-4.35 E-01	-9.75 E-02	-2.34 E-01	1.73 E+00	-3.63 E-01	-1.78 E-01	-1.30 E+00	-1.79 E+01
Kurtosis	1340.82	95.95	65.18	111.74	13.25	76.32	3.23	0.25	10.33	0.24

Fig. 9. Statistical characteristics of the three clusters

In this case we note that the value of the mean changes most for BirthYear. Cluster 1 has a very different value for the mean for DeathCause than the other two clusters. The decoded values of the mode for BirthYear and DeathCause are "2008" and Heart stroke, for cluster 2 "2008" and "Unknown", and for cluster 2 "1990" and "Unknown". Additionally we observe also significant changes in the mean for AilmentGroup. When decoding the values of the mode in each cluster we get that for cluster 1 the mode is Trombosis (in effect a heart condition), for cluster 2 it is Diabetes type 2 and for cluster 3 it is Diabetes type 1.

4 Discussion and Perspectives

We have shown that we are able to find meaningful results by applying numerically oriented non-supervised clustering algorithms to categorical data by properly encoding the instances of the categories. We were able to determine the number of clusters arising from the data encoded according to our algorithm and, furthermore, to interpret the clusters in a meaningful way. When comparing the clusters determined by our method to those of human experts we found some coincidences. However, some of our conclusions do not match those of the experts.

Rather than assuming that this is a limitation of our method, we would prefer to suggest that machine learning techniques such as the one described, yield a broader scope of interpretation because they are not marred by limitations of processing capabilities which are evident in any human attempt to encompass a large set of data.

At any rate, the proposed encoding does allow us to tackle complex problems without the limitations derived from the non-numerical characteristics of the data. Much work remains to be done, but we are confident that these are the first of a series of significant applications.

References

1. Agresti, A.: Categorical Data Analysis, 2nd edn. Wiley Series in Probability and Statistics. Wiley- Interscience (2002)
2. Barbará, D., Li, Y., Couto, J.: Coolcat: an entropy-based algorithm for categorical clustering. In: CIKM 2002: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 582–589. ACM, New York (2002)
3. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: A comparative evaluation. In: SDM, pp. 243–254 (2008)
4. Cesario, E., Manco, G., Ortale, R.: Top-down parameter-free clustering of high-dimensional categorical data. IEEE Trans. on Knowl. and Data Eng. 19(12), 1607–1624 (2007)
5. Chandola, V., Boriah, S., Kumar, V.: A framework for exploring categorical data. In: SDM, pp. 185–196 (2009)
6. Chang, C.-H., Ding, Z.-K.: Categorical data visualization and clustering using subjective factors. Data Knowl. Eng. 53(3), 243–262 (2005)
7. Ganti, V., Gehrke, J., Ramakrishnan, R.: Cactus—clustering categorical data using summaries. In: KDD 1999: Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 73–83. ACM, New York (1999)
8. Gibson, D., Kleinberg, J., Raghavan, P.: Clustering categorical data: an approach based on dynamical systems. The VLDB Journal 8(3-4), 222–236 (2000)
9. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. In: ICDE Conference, pp. 512–521 (1999)
10. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 1st edn. Morgan Kaufmann, San Francisco (2001)
11. Hsu, C.-C., Wang, S.-H.: An integrated framework for visualized and exploratory pattern discovery in mixed data. IEEE Trans. on Knowl. and Data Eng. 18(2), 161–173 (2006)
12. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2(3), 283–304 (1998)
13. Lee, J., Lee, Y.-J., Park, M.: Clustering with Domain Value Dissimilarity for Categorical Data. In: Perner, P. (ed.) ICDM 2009. LNCS, vol. 5633, pp. 310–324. Springer, Heidelberg (2009)

14. Johansson, S., Jern, M., Johansson, J.: Interactive quantification of categorical variables in mixed data sets. In: IV 2008: Proceedings of the 2008 12th International Conference Information Visualisation, pp. 3–10. IEEE Computer Society, Washington, DC, USA (2008)
15. Koyuturk, M., Grama, A., Ramakrishnan, N.: Compression, clustering, and pattern discovery in very high-dimensional discrete-attribute data sets. IEEE Trans. on Knowl. and Data Eng. 17(4), 447–461 (2005)
16. Wang, K., Xu, C., Liu, B.: Clustering transactions using large items. In: ACM CIKM Conference, pp. 483–490 (1999)
17. Yan, H., Chen, K., Liu, L.: Efficiently clustering transactional data with weighted coverage density. In: CIKM 2006: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 367–376. ACM, New York (2006)
18. Yang, Y., Guan, X., You, J.: Clope: a fast and effective clustering algorithm for transactional data. In: KDD 2002: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 682–687. ACM, New York (2002)
19. Haykin, S.: Neural networks: A comprehensive foundation. MacMillan (1994)
20. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. J. Intell. Inf. Syst. 17(2-3), 107–145 (2001)
21. Jenssen, R., Hild, K.E., Erdogmus, D., Principe, J.C., Eltoft, T.: Clustering using Renyi's entropy. In: Proceedings of the International Joint Conference on Neural Networks 2003, vol. 1, pp. 523–528 (2003)
22. Lee, Y., Choi, S.: Minimum entropy, k-means, spectral clustering. In: Proceedings IEEE International Joint Conference on Neural Networks, 2004, vol. 1 (2005)
23. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. Scientific American (July 1949)
24. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clustering's comparison: is a correction for chance necessary? In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1073–1080 (2009)
25. Teuvo, K.: Self-organizing maps. Springer-Verlag, New York, Inc., Secaucus (1999)
26. `http://udel.edu/~mcdonald/statspearman.html` (August 26, 2011)
27. `http://www.mei.org.uk/files/pdf/Spearmanrcc.pdf` (September 9, 2011)

Regional Flood Frequency Estimation for the Mexican Mixteca Region by Clustering Techniques

Felix Emilio Luis-Pérez[1], Raúl Cruz-Barbosa[1], and Gabriela Álvarez-Olguin[2]

[1] Computer Science Institute
{eluis,rcruz}@mixteco.utm.mx
[2] Hydrology Institute
Universidad Tecnológica de la Mixteca
69000, Huajuapan, Oaxaca, México
galvarez@mixteco.utm.mx

Abstract. Regionalization methods can help to transfer information from gauged catchments to ungauged river basins. Finding homogeneous regions is crucial for regional flood frequency estimation at ungauged sites. As it is the case for the Mexican Mixteca region site, where actually only one gauging station is working at present. One way of delineate these homogeneous watersheds into natural groups is by clustering techniques. In this paper, two different clustering approaches are used and compared for the delineation of homogeneous regions. The first one is the hierarchical clustering approach, which is widely used for regionalization studies. The second one is the Fuzzy C-Means technique which allow a station belong, at different grades, to several regions. The optimal number of regions is based on fuzzy cluster validation measures. The experimental results of both approaches are similar which confirm the delineated homogeneous region for this case study. Finally, the stepwise regression model using the forward selection approach is applied for the flood frequency estimation in each found homogeneous region.

Keywords: Regionalization, Fuzzy C-Means, Hierarchical Clustering, Stepwise Regression Model, Mexican Mixteca Region.

1 Introduction

In areas where water is insufficient to meet the demands of human activities, the evaluation of water availability is a key factor in creating efficient strategies for its optimal use. An example of these areas is the Mexican Mixteca region site, where the gauging stations have declined due to high maintenance costs and the continuing deterioration of them. According to [1], since 1940, 13 gauging stations were installed in this region and only one is in operation at present.

In this kind of areas, regionalization methods can help to transfer information from gauged catchments to ungauged river basins [2]. This technique can be applied in design of water control structures, economic evaluation of flood

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 249–260, 2011.

protection projects, land use planning and management, and other hydrologic studies.

Finding homogeneous regions is crucial for regional flood frequency estimation at ungauged sites. Several approaches have been adapted to the purpose of homogeneous region delineation. Among the most prominent approaches for these task, we can identify canonical correlation analysis [3], and cluster analysis [4]. Some of the mainly used techniques in cluster analysis for homogeneous region delineation is hierarchical and fuzzy clustering.

The main objective of this paper is to estimate the regional flood frequency for the Mexican Mixteca river basin in the state of Oaxaca, Mexico. For this purpose, hierarchical and fuzzy clustering techniques are used for homogeneous region delineation. Further, the corresponding results of these approaches are compared and interpreted. The historical record of monthly flows from 1971 to 1977 of 10 hydrological stations are used for this analysis. Finally, the stepwise regression model using the forward selection approach is applied for the flood frequency estimation in each found homogeneous region.

2 Related Work

There are several ways to find homogeneous regions for regionalization tasks. Two main techniques used to delineate homogeneous regions are canonical correlation and clustering analysis. In [5] and [6], canonical correlation analysis is applied to several river basins located at northAmerica (Canada and U.S.A.).

In contrast, in [7] authors use the ward linkage clustering, the Fuzzy C-Means method and a Kohonen neural network in the southeast of China to delineate homogeneous regions. Using a different approach, the k-means method is used in a study with selected catchments in Great Britain [8].

In Mexico, an important research for regional estimation was conducted in 2008 [9]. Four approaches for the delineation of homogeneous regions are used in this study: The hierarchical clustering analysis, the canonical correlation analysis, a revised version of the canonical correlation analysis, and the canonical kriging. Following this fashion, in some way, a first hydrological regionalization study for the Mexican Mixteca Region was carried out in [10]. The delineation of homogeneous regions was determined by hierarchical clustering methods and the Andrews technique. Here, a different and a greater number of gauging stations than in this paper were analyzed.

On the other hand, with regard to the question whether to use linear or nonlinear regression models as a regional estimation method, a comparison between linear regression models and artificial neural nets for linking model parameters to physical catchment descriptors is shown in [11]. They conclude that the linear regression model is the most commonly used tool, however artificial neural networks are a useful alternative if the relationship between model parameters and catchment descriptors is previously known to be nonlinear.

3 Regional Flood Frequency Analysis

According to [9] the regional estimation methodologies involve two main steps: the identification of groups of hydrologically homogeneous basins or "homogeneous regions" and the application of a regional estimation method within each delineated homogeneous region. In this study, the relationship between the flood frequency and the climatic and physiographic variables is unknown, therefore it was applied, for regionalization purpose, multiple linear regression analysis.

In the context of regional flood frequency, homogeneous region can be defined as fixed regions (geographically contiguous or not-contiguous regions), or as hydrological neighborhoods. The delineation of homogeneous hydrologic regions is the most difficult step and one of the most serious obstacles for a successful regional solution [12]. One way of delineate the homogeneous regions into natural groups is by clustering techniques.

3.1 Clustering

Cluster analysis is the organization of a collection of patterns into clusters based on similarity [4]. When it is applied to a set of heterogeneous items, it identifies homogeneous subgroups according to proximity between items in a data set.

Clustering methods can be divided into hierarchical and non-hierarchical [13]. The former constructs a hierarchical tree of nested data partitions. Any section of the tree at a certain level produces a specific partition of the data. These methods can be divided into agglomerative and divisive depending on how they build the clustering tree. Non-hierarchical clustering methods, despite their variety, they have the characteristic that all of them require the number of clusters to be partitioned.

For our case study, we focus on hierarchical and Fuzzy C-Means clustering as a tool for the delineation of homogeneous regions. These methods have been successfully applied in this kind of problem [9,14,15] .

Hierarchical Clustering. These algorithms are characterized by having a tree shaped structure, which is commonly called dendrogram. Here, each level is a possible clustering of objects in the data collection [4]. Each vertex or node of the tree represents a group of objects, and the tree root contains all items in the collection, forming a single group.

There are two basic approaches for hierarchical clustering: agglomerative and divisive. Agglomerative approach starts with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. Divisive clustering starts with one cluster (all points included) and, at each step, split a cluster until only singleton clusters of individual points remain.

In hierarchical clustering, the obtained clusters depend on the considered distance criterion. That is, clustering depends on the (dis)similarity criteria used to group the data. The most frequently used similarity measure is the Euclidean distance. It can also be used other similarity measures such as the Manhattan or Chebyshev distance.

Another issue to be considered in this kind of clustering is the linkage function. It determines the homogeneity degree that may exist between two sets of observations. The most common linkage function are: average linkage, centroid linkage and ward linkage.

Fuzzy C-Means Clustering. The concept of fuzzy sets arises when modeling of systems is impossible by the mathematical precision of classical methods, i.e., when data to be analyzed have some uncertainty in their values, or the data do not have specific value [16].

The Fuzzy C-Means (FCM) algorithm [17] is one of the most widely used methods in fuzzy clustering. It is based on the concept of fuzzy c-partition, introduced by [18]. The aim of the of FCM algorithm is to find an optimal fuzzy c-partition and corresponding prototypes minimizing the objective function.

$$J(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m ||x_k - v_i||^2 \tag{1}$$

where, $X = \{x_1, ..., x_n\}$ is a data set , each data point x_k is an input vector, $V = (v_1, v_2, ..., v_c)$ is a matrix of unknown cluster centers, U is a membership matrix, u_{ik} is the membership value of x_k in cluster i $(i = 1, ...c)$, and the weighting exponent m in $[1, \infty]$ is a constant that influences the membership values.

In each iteration, it is necessary to update the cluster centroids using Eq. 2, and given the new centroids, also it is necessary to update membership values using Eq. 3. The stop condition of the algorithm is using the error between the previous and current membership values.

$$\hat{c}_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m} \tag{2}$$

$$\hat{u}_{ik} = \left[\sum_{j=1}^{c}\left(\frac{||x_k - v_i||^2}{||x_k - v_j||^2}\right)^{\frac{2}{m-1}}\right]^{-1} \tag{3}$$

Cluster validity indices have been extensively used to determine optimal number of clusters c in a data set. In this study, four cluster validity measures namely, Fuzzy Partition Coefficient (V_{PC}), Fuzzy Partition Entropy (V_{PE}), Fuzziness Performance Index (FPI) and Normalized Classification Entropy (NCE) are computed for different values of both c and U. These indices can help to derive hydrologically homogeneous regions. Furthermore, these indices, which are not directly related to properties of the data, have been previously used in hydrological studies [14].

The validity indices V_{PC} and V_{PE} proposed by [19], and the indices FPI and NCE introduced by [20] are defined as:

$$V_{PC}(U) = \frac{1}{n}\sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^2 \tag{4}$$

$$V_{PE}(U) = -\frac{1}{n}\left[\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik} log(u_{ik})\right] \tag{5}$$

$$FPI(U) = 1 - \frac{c\, x\, V_{PC}(U) - 1}{c - 1} \tag{6}$$

$$NCE(U) = \frac{V_{PE}(U)}{log(c)} \tag{7}$$

The optimal partition corresponds to a maximum value of V_{PC} (or minimum value of V_{PE}, FPI and NCE), which implies minimum overlap between cluster elements.

3.2 Multiple Linear Regression

The Multiple Linear Regression (MLR) method is used to model the linear relationship between a dependent variable and two or more independent variables. The dependent variable is sometimes called the predictand, and the independent variables the predictors [21]. MLR is based on least squares: the model is fitted such that the sum-of-squares of the observed differences and predicted values is minimized. The model expresses the value of a predictand variable as a linear function of one or more predictor variables and an error term as follows:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + ... + \beta_k x_{i,k} + \varepsilon \tag{8}$$

Where $x_{i,k}$ is the value of the $k - th$ predictor for the $i - th$ observation, β_0 is a regression constant, β_k is the $k - th$ predictor coefficient, y_i is the predictand for the $i - th$ observation and ε the error term.

The Eq. 8 is estimated by least squares, which yields the estimation of β_k and y_i parameters.

In many cases, MLR assumes that all predictors included in the model are important. However, in practical problems the analyst has a set of candidate variables, which should determine the true subset of predictors to be used in the model. The definition of an appropriate subset of predictors for the model is what is called Stepwise Regression [21].

4 Experiments

4.1 Experimental Design and Settings

The main objectives of the experiment are: to obtain the homogeneous regions for the Mexican Mixteca region and to estimate the regional flood frequency for each previously found region. Firstly, the delineation of homogeneous regions using the hierarchical technique and Fuzzy C-Means approach is carried out. Secondly, the regionalization model for each previously found cluster is achieved by using stepwise regression approach.

Table 1. Gauging hydrometric stations used in the study, with historical record of monthly flows from 1971 to 1977

Station	Basin	Code	Water Region	State
Apoala	Papaloapan	28082	Papaloapan	Oaxaca
Axusco	Salado	28102	Papaloapan	Oaxaca
Ixcamilca	Mezcala	18432	Balsas	Puebla
Ixtayutla	Verde	20021	Balsas	Oaxaca
Las Juntas	Ometepec	20025	Costa Chica	Guerrero
Nusutia	Yolotepec	20041	Costa Chica	Oaxaca
San Mateo	Mixteco	18352	Balsas	Oaxaca
Tamazulapan	Salado	18433	Balsas	Oaxaca
Teponahuazo	Grande	18342	Balsas	Guerrero
Xiquila	Papaloapan	28072	Papaloapan	Oaxaca

The data set was obtained from ten river gauging stations, as shown in Table 1. The historical records of monthly flows from 1971 to 1977 were used for each station and these were taken from Sistema de Información de Aguas Superficiales edited by Instituto Mexicano de Tecnología del Agua (IMTA) [22]. Only these gauging hydrometric station are used because they have the largest historical measurements.

Once we have selected the gauging stations for the study, the quality of hydrometric data was checked by applying the Wald-Wolfowitz test for independence, the Mann-Whitney test for homogeneity, and the Grubbs-Beck test for outliers (using 5% of significance level). As a result of these tests we remove four outliers from Apoala station, three from Axusco, three from Nusutia, two from Ixtayutla, two from Tamazulapan, two from Teponahuazo and two from Xiquila.

Also, the data were standardized because of scale problems, using the following expression:

$$y_{i,j} = \frac{x_{i,j} - \bar{x}_i}{S_x} \tag{9}$$

where $x_{i,j}$ represents the value of the $j-th$ observation of the $i-th$ variable, $\bar{x}_i$ is the average of the variable i, S_x represent the standard deviation, and $y_{i,j}$ is the representation of the $j-th$ observation of the $i-th$ transformed variable.

4.2 Experimental Results and Discussion

As we explained in section 4.1, first the clustering results are presented in order to show the homogeneous regions for this case study. In the second stage multiple linear regression approach is used for regional estimation.

The application of the hierarchical cluster analysis technique leads to the dendrograms shown in Figs. 1 - 3. In each case we can identify two groups, each one representing a homogeneous region. The first region includes Tamazulapan, Xiquila, Axusco and Apoala stations, and the second region includes San-Mateo, Teponahuazo, Ixcamilca, Ixtayutla, Las-Juntas and Nusutia.

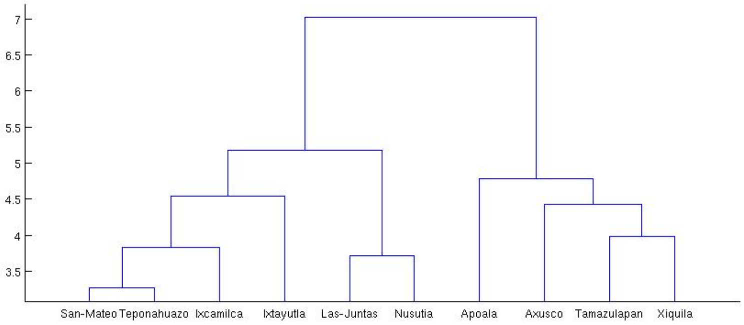


Fig. 1. Hierarchical clustering results using average linkage

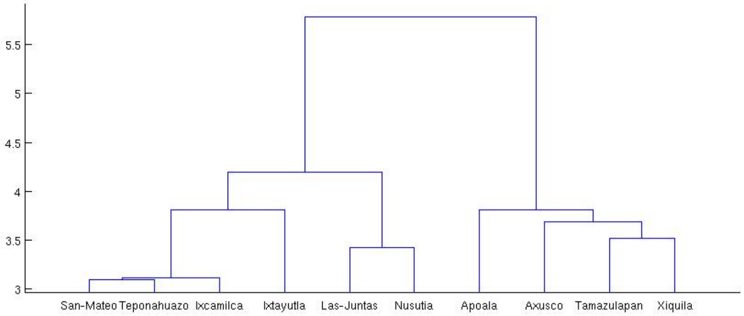


Fig. 2. Hierarchical clustering results using centroid linkage

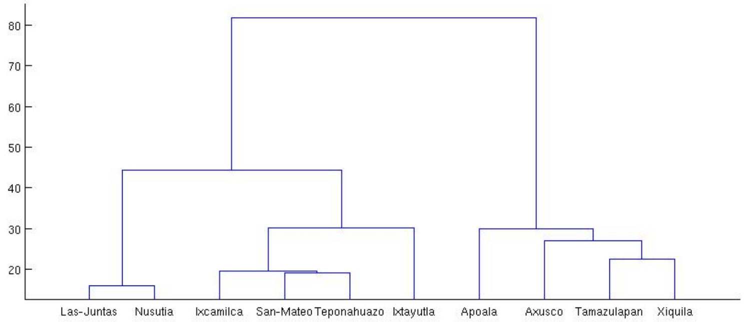


Fig. 3. Hierarchical clustering results using ward linkage

It can be observed that for the average and centroid linkage, a possible cutting distance is 5.5, whereas the ward linkage it is 50. Overall, the order in the subgroups is maintained, except for the ward linkage where Las-Juntas and Nusutia station form the first group.

Table 2. Fuzzy C-Means clustering results using two clusters

Cluster 1	Cluster 2
Apoala	Ixcamilca
Axusco	Ixtayutla
Tamazulapan	Las-Juntas
Xiquila	Nusutia
	San-Mateo
	Teponahuazo

Table 3. Fuzzy C-Means clustering results using three clusters

Cluster 1	Cluster 2	Cluster 3
Ixtayutla	Apoala	Ixcamilca
Las-Juntas	Axusco	San-Mateo
Nusutia	Tamazulapan	Teponahuazo
	Xiquila	

Table 4. Fuzzy C-Means clustering results using four clusters

Cluster 1	Cluster 2	Cluster 3	Cluster 4
San-Mateo	Apoala	Ixcamilca	Las-Juntas
Teponahuazo	Axusco	Ixtayutla	Nusutia
	Tamazulapan		
	Xiquila		

For the Fuzzy C-Means clustering results a defuzzifier is used to convert the obtained fuzzy set values into crisp values. A usual defuzzifier is the maximum-membership method used in [15], which for each instance x_k it takes the largest element in the $k-th$ column of the membership matrix U and assigns a new grade of membership value of one to it and the other column elements are assigned a membership grade of zero. That is,

$$u_{ik} = max(u_{jk}) = 1, \quad \forall\, 1 \leq j \leq c; \qquad u_{ik} = 0 \quad \forall\, i \neq j \tag{10}$$

The delineation of homogeneous regions using the Fuzzy C-Means algorithm was computed for four diferent cases. In the first case, the number of predefined clusters was two, and the obtained regions are shown in Table 2. These results coincide with the hierarchical clustering results.

For the remaining cases, we defined three, four and five fuzzy clusters, and the distribution of the gauged stations are presented in Tables 3 to 5, respectively.

Overall, both kind of clustering are consistent. In particular, the group formed by Apoala, Axusco, Tamazulapan and Xiquila stations is maintained through different clustering experiments, as show in Tables 2 - 4. The other groups from Tables 3 to 5, are consistent with the subgroups formed by using hierarchical clustering.

Table 5. Fuzzy C-Means clustering results using five clusters

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Axusco Tamazulapan Xiquila	Apoala	Ixcamilca Teponahuazo	Ixtayutla San-Mateo	Las-Juntas Nusutia

Table 6. Cluster validity measurement results

	Number of clusters			
Index	2	3	4	5
V_{PC}	**0.693**	0.566	0.419	0.435
V_{PE}	**0.208**	0.330	0.466	0.491
FPI	**0.612**	0.649	0.773	0.705
NCE	0.693	**0.692**	0.774	0.703

The optimal number of clusters for a data set can be identified by applying fuzzy cluster validation measures on the partitions obtained from the second level of the Fuzzy C-Means method. Some of these measures are the Fuzzy Partition Coefficient V_{PC}, the Fuzzy Partition Entropy V_{PE}, the Fuzziness Performance Index FPI and the Normalized Classification Entropy NCE.

The corresponding results of applying these measures are shown in Table 6. Here V_{PC}, V_{PE} and FPI which have been used in the hydrologic literature [23], clearly suggest two clusters as the best partition, irrespective of the structure in the data being analyzed. Although the NCE measure weakly suggest three clusters as the best partition, this result is very similar for two clusters.

After the homogeneous regions were obtained, multiple regression approach is used for regional estimation. For the inherent basins in the ten hydrometric stations, four climatic variables and ten physiographic variables were quantified, all potentially adequate in flow frequency estimation. The independent variables used in the regressive model are, monthly mean precipitation, main channel length, forest covering, temperature, annual mean precipitation, basin area, drainage density, basin mean elevation, soil runoff coefficient, maximum and minimum basin elevation, latitude, longitude and the annual maximum rainfall in 24 hours with return period of 2 years. Consequently the dependent variables are the maximum flow Q_{max}, minimum flow Q_{min} and mean flow Q_{mean}.

The four climatic variables, monthly mean precipitation, annual mean precipitation, temperature and the annual maximum rainfall in 24 hours with return period of 2 years were obtained of the daily series of rain and temperature of Extractor Rápido de Información Climatológica V3 designed by CONAGUA [1]. The physiographic variables were estimated from images of the LANDSAT [24] satellite in 1979. These variables were processed by Sistema de Procesamiento de Información Topográfica of INEGI [25].

When multiple regression using all independent variables is applied, the resulting model is very large and unusable because it is very difficult to get the values for all the involved variables. Thus, the stepwise regression approach is used, specifically forward selection. Applying this method using 5% of significant level to the first cluster determined by clustering algorithms, we found the regression models shown in Eq. 11 to Eq. 13.

$$Q_{max} = -133.64 + 0.57x_1 + 1.69x_2 + 0.109x_3 \tag{11}$$

$$Q_{min} = -5.73 + 0.053x_2 + 0.00725x_3 + 0.00227x_1 \tag{12}$$

$$Q_{mean} = -133.68 + 0.027x_1 + 0.123x_2 + 0.0159x_3 \tag{13}$$

where x_1 is the monthly mean precipitation, x_2 is the main channel length and x_3 is annual mean precipitation. For the maximum flood, the coefficient of multiple determination(R^2) is 0.46, for the minimum flood is 0.48, and for the mean flood is 0.46. This means that the proposed models have a good tendency to describe the variability of the data set.

It can be observed that these regression models include the same independent variables, however, these do not have the same importance for each model.

On the other hand, the regression models for the second cluster using 5% of significant level are shown in Eq. 14 to Eq. 16

$$Q_{max} = -112.88 + 1.58x_1 + 1.87x_2 \tag{14}$$

$$\begin{aligned} Q_{min} = 158.62 - 0.0671x_3 + 0.098x_1 \\ +0.02053x_4 - 1.36x_2 - 0.033x_5 \end{aligned} \tag{15}$$

$$\begin{aligned} Q_{mean} = -106.25 + 0.334x_1 + 0.0351x_3 \\ +0.0180x_4 + 1.70x_6 - 0.0192x_7 \end{aligned} \tag{16}$$

where x_1 is the monthly mean precipitation, x_2 is the main channel length, x_3 is the minimum elevation, x_4 is the basin area, x_5 is the annual mean precipitation, x_6 represent the soil runoff coefficient and x_7 is mean elevation basin. In this case, the coefficient of multiple determination for the maximum flood is 0.40, for the minimum flood is 0.38, and for the mean flood is 0.5. These results show that the most reliable model is for the mean flood, which can describe an important variability of the data.

5 Conclusion

Regionalization methods are very useful for regional flood frequency estimation, mainly, at ungauged sites. In this paper, the Mexican Mixteca Region is analyzed for regionalization studies. In a first stage, the homogeneous watersheds are found by clustering techniques. The Hierarchical and Fuzzy C-Means clustering are applied to ten gauging station data. Experimental results have shown that this data set can be grouped into two homogeneous regions, which is confirmed by both applied kinds of clustering.

The stepwise regression model using the forward selection approach and 5% of significant level is applied for the flood frequency estimation in the second stage of this study. The obtained models have shown that only the monthly mean precipitation, the main channel length and the annual mean precipitation variables are needed to estimate the maximum, minimum and mean flow in the first found homogeneous region and for the second region the monthly mean precipitation, the main channel length, the minimum elevation, the basin area, the annual mean precipitation, the soil runoff coefficient and the mean elevation basin variables are required. Overall, few variables are needed to estimate the maximum, minimum and mean flow in each region.

Further research should include more types of regression models, as well as a comparison of them in terms of the number and the importance of the used variables.

References

1. CONAGUA: Comisión nacional de agua. Dirección Técnica del Organismo de Cuenca Balsas, Oaxaca, Mexico (August 20, 2010), http://www.cna.gob.mx
2. Nathan, R., McMahon, T.: Identification of homogeneous regions for the purposes of regionalization. Journal of Hydrology 121, 217–238 (1990)
3. Ouarda, T., Girard, C., Cavadias, G., Bobée, B.: Regional flood frequency estimation with canonical correlation analysis. Journal of Hydrology 254, 157–173 (2001)
4. Jain, A., Murty, M., Flinn, P.: Data clustering: A review. ACM Computing Surveys 31(3) (1999)
5. Shih-Min, C., Ting-Kuei, T., Stephan, J.: Hydrologic regionalization of watersheds. ii: Applications. Journal of water resources planning and management 128(1) (2002)
6. Leclerc, M., Ouarda, T.: Non-stationary regional flood frequency analysis at ungauged sites. Journal of Hydrology 343, 254–265 (2007)
7. Jingyi, Z., Hall, M.: Regional flood frequency analysis for the gan-ming river basin in china. Journal of Hydrology 296, 98–117 (2004)
8. Chang, S., Donald, H.: Spatial patterns of homogeneous pooling groups for flood frequency analysis. Hydrological Sciences Journal 48(4), 601–618 (2003)
9. Ouarda, T., Ba, K., Diaz-Delgado, C., Carsteanu, A., Chokmani, K., Gingras, H., Quentin, E., Trujillo, E., Bobée, B.: Intercomparison of regional flood frequency estimation methods at ungauged sites for a mexican case study. Journal of Hydrology 348, 40–58 (2008)
10. Hotait-Salas, N.: Propuesta de regionalización hidrológica de la mixteca oaxaqueña, méxico, a través de análisis multivariante. Universidad Politécnica de Madrid, Tesis de Licenciatura (2008)
11. Heuvelmans, G., Muys, B., Feyen, J.: Regionalisation of the parameters of a hydrological model: Comparison of linear regression models with artificial neural nets. Journal of Hidrology 319, 245–265 (2006)

12. Smithers, J., Schulze, R.: A methodology for the estimation of short duration design storms in south africa using a regional approach based on l-moments. Journal of Hydrology 24, 42–52 (2001)
13. Downs, G.M., Barnard, J.M.: Clustering methods and their uses in computational chemistry. In: Lipkowitz, K.B., Boyd, D.B. (eds.) Reviews in Computational Chemistry, Hoboken, New Jersey, USA, vol. 18 (2003)
14. Guiler, C., Thine, G.: Delineation of hidrochemical facies distribution in a regional groundwater system by means of fuzzy c-means clustering. Water Resources Research (40) (2004)
15. Srinivas, V.V., Tripathi, S., Rao, R., Govindaraju, R.: Regional flood frequency analysis by combining self-organizing feature map and fuzzy clustering. Journal of hidrology 348, 146–166 (2008)
16. Zheru, C., Hong, Y., Tuan, P.: Fuzzy Algorithms: with aplications to image procesing and pattern recognition, vol. 10. World Scientific Publishing, Singapore (1996)
17. Bezdek, J.C.: Pattern recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
18. Ruspini, E.: A new approach to clustering. Information and control 15, 22–32 (1969)
19. Bezdek, J.: Cluster validity with fuzzy sets. Journal of cybernetics 3(3), 58–72 (1974)
20. Roubens, M.: Fuzzy clustering algorithms and their cluster validity. European Journal of Operations Research (10), 294–301 (1982)
21. Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to Linear Regression Analysis, 3rd edn. Wiley-Interscience, New York (2001)
22. IMTA: Instituto mexicano de tecnología del agua. Sistema de Información de Aguas Superficiales (June 13, 2011), http://www.imta.gob.mx
23. Hall, M., Mins, A.: The classification of hydrologically homogeneous region. Hydrological Science Journal 44, 693–704 (1999)
24. LANDSAT: The landsat program. National Aeronautics and Space Administration (June 13, 2011), http://landsat.gsfc.nasa.gov
25. INEGI: Instituto nacional de estadística, geografía e informática. Sistema de Procesamiento de Información Topográfica (June 13, 2011), http://www.inegi.org.mx

Border Samples Detection for Data Mining Applications Using Non Convex Hulls

Asdrúbal López Chau[1,3], Xiaoou Li[1], Wen Yu,[2],
Jair Cervantes[3], and Pedro Mejía-Álvarez[1]

[1] Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico
achau@computacion.cs.cinvestav.mx, {lixo,pmalavrez}@cs.cinvestav.mx
[2] Automatic Control Department, CINVESTAV-IPN, Mexico City, Mexico
yuw@ctrl.cinvestav.mx
[3] Graduate and Researh, Autonomous University of Mexico State,Texcoco Mexico
chazarra17@gmail.com

Abstract. Border points are those instances located at the outer margin of dense clusters of samples. The detection is important in many areas such as data mining, image processing, robotics, geographic information systems and pattern recognition. In this paper we propose a novel method to detect border samples. The proposed method makes use of a discretization and works on partitions of the set of points. Then the border samples are detected by applying an algorithm similar to the presented in reference [8] on the sides of convex hulls. We apply the novel algorithm on classification task of data mining; experimental results show the effectiveness of our method.

Keywords: Data mining, border samples, convex hull, non-convex hull, support vector machines.

1 Introduction

Geometric notion of *shape* has no associated a formal meaning[1], however intuitively the shape of a set of points should be determined by the borders or boundary samples of the set. The boundary points are very important for several applications such as robotics [2], computer vision [3], data mining and pattern recognition [4]. Topologically, the boundary of a set of points is the closure of it and defines its shape[3]. The boundary does not belong to the interior of the shape.

The computation of border samples that better represent the shape of set of points has been investigated for a long time. One of the first algorithms to compute it is the convex hull (CH). The CH of a set of points is the minimum convex set that contains all points of the set. A problem with CH is that in many cases, it can not represent the shape of a set, i.e., for set of points having interior "corners" or concavities the CH ommits the points that determine the border of those areas. An example of this can be seen in Fig. 1.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 261–272, 2011.

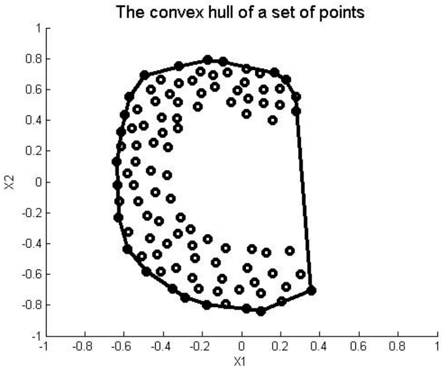


Fig. 1. Convex hull can not represent exactly **the borders** of all sets of points

In order to better characterize the region occupied for a set of points, some proposals have been presented : alpha shapes, conformal alpha shapes, concave hull algorithm and Delaunay-based methods.

In [5] the alpha-shapes as a generalization of the convex hull was presented. Alpha shapes seem to capture the intuitive notions of "fine shape" and "crude shape" of point sets. This algorithm was extended to more than two dimensions in [1]. In [6] is proposed a solution to compute the “footprint” of a set of points. Different from geometric approach was proposed in [7], were the boundary points are recovered based on the observation that they tend to have fewer reverse k-nearest neighbors. An algorithm based on Jarvis march was presented in [8], the algorithm is able to efficiently compute the boundary of a set of points in two dimensions. A problem detected with the algorithm in [8] is that although it can effectively work in almost all scenarios, in some cases it produces a set of elements that does not contain all the samples in a given data set, this is specially notorious if the distribution of samples is not uniform, i.e., if there are “empty” zones, another detail occurs if there are several clusters of points, the algorithm does not compute all border samples.

In this paper we introduce an algorithm to compute border samples. The algorithm is based on the presented in [8] but with the following differences: The algorithm was modified to be able to compute all extreme points, the original algorithm sometimes ignores certain points and the vertexes of convex hull are not included as part of the solution. Instead of using the k nearest neighbors of a point p_i we use the points that are within a hyper-box centered in p_i, this makes the algorithm slightly faster than the original one if the points are previously sorted. The novel algorithm was extended for higher dimensions using a clustering strategy. Finally we use a discretization step and work with groups of adjacent cells from where the border samples are detected.

The rest of the paper is organized as follows. In the section 2 definitions about convexity, convex and non convex hulls are explained. The notion of border samples is also shown. Then in section 3 three useful properties to compute border samples of a set of points are shown, and proposed algorithms that accomplish the properties are explained. In section 4 the method is applied as a pre-processing step in classification task using Support Vector Machine (SVM) as an application of the algorithms to data mining. The results show the effectiveness of the proposed method. Conclusions and future work are in last part of this paper.

2 Border Points, Convex Hull and Non-convex Hull

The *boundary points (or border points)* of a data set are defined in [7] as those ones that satisfy the following two properties: Given a set of points $P = \{p \in R^n\}$, a $p \in P$ is a border one if

1. It is within a dense region R and
2. $\exists$ region R$^{'}$ near of p such that Density(R $^{'}$) $\gg$ Density(R).

The *convex hull* $\mathcal{CH}$ of a data set X is mathematically defined as in equation (1) and there are several algorithms to compute it [9]: brute force ($\mathcal{O}(n^3)$), Grahamś scan($\mathcal{O}(n \log n)$), divide and conquer $\mathcal{O}(n \log n)$, quick hull (average case $\mathcal{O}(n \log n)$, Jarviś march and Chan's algorithm ($\mathcal{O}(n \log h)$.

$$\mathcal{CH}(X) \left\{ w : w = \sum_{i=1}^{n} a_i x_i, a_i \geq 0, \sum_{i=1}^{n} a_i = 1, x_i \in X \right\} \tag{1}$$

Extreme points are the vertexes of the convex hull at which the interior angle is strictly convex[10]. However as stated before and exemplified in figure 1, $\mathcal{CH}(X)$ can not always capture all border samples of X. Another detail relates to the use of $\mathcal{CH}$ for capturing the border samples occurs when the set of points form several groups or clusters, only extreme borders are computed and outer borders are omitted. For cases like this, the border samples $\mathcal{B}(\cdot)$ usually should define a non-convex set. A convex set is defined as follows[11]: A set S in R^n is said to be convex if for each

$$x_1, x_2 \in S, \alpha x_1 + (1 - \alpha) x_2 \textit{ belongs to } S \tag{2}$$
$$for\ \alpha \in (0, 1).$$

Any set $S^{'}$ that does not hold equation (2) is called a *non-convex.*

We want to compute $\mathcal{B}(X)$ which *envelopes* a set of points, i.e., $\mathcal{B}(X)$ is formed with the borders of X. Because a data set is in general non-convex, we call $B(X)$ *non-convex hull.* The terms non-convex hull and border samples will be used interchangeably in this work.

Although the $\mathcal{CH}(P)$ is unique for each set of points P, the same does not occur with $\mathcal{B}(\mathcal{P})$, there can be more than one valid set of points that define the border for the given P. An example of this is shown in figure 2. The difference of between two border sets $\mathcal{B}(\mathcal{P})$ and $\mathcal{B}(\mathcal{P}')$ is due to the size of each one, which

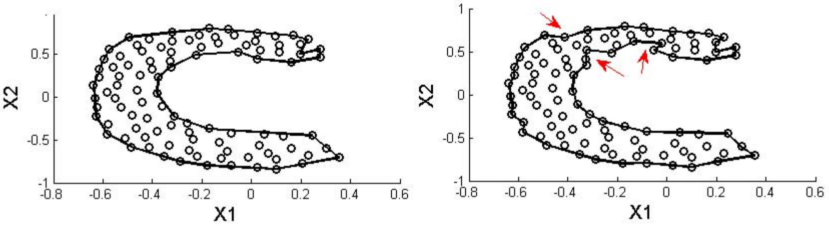


Fig. 2. Two different non-convex hulls for the same set of points. The arrows show some differences.

in turn is related with the degree of detail of the shape. A non-convex hull with a small number of points is faster to compute, but contain less information and vice-verse. This flexibility can be exploited depending on application.

The minimum and maximum size ($|\cdot|$) of a $\mathcal{B}(\mathcal{P})$ for a given set of points P is determined by (3) and (4).

$$min\ |\mathcal{B}(\mathcal{P})| = |\mathcal{CH}(P)|\ \ \forall\ \mathcal{B}(P). \tag{3}$$

$$max|\mathcal{B}(\mathcal{P})| = |P|\ \ \forall\ \mathcal{B}(P). \tag{4}$$

The (3) and (4) are directly derived, the former is from definition of $\mathcal{CH}$, whereas the second happens when $\mathcal{B}(P)$ contains all the points.

Let be P $=\{p \in R^n\}$ and $P^{'}$ a discretization of P by using a grid method. Let be y_i a cell of the grid and let be Y_i a group of adjacent cells with $\bigcap_i Y_i = \varnothing$ and $\bigcup_i Y_i = P^{'}$. The following three properties contribute to detect the border samples of $P^{'}$.

1. $\forall\ \mathcal{B}(P^{'}),\ \mathcal{B}(P^{'}) \supset \mathcal{CH}(P^{'})$.
2. $\bigcup_i \mathcal{B}(Y_i) \supset \mathcal{B}(P^{'})$.
3. *Vertexes of* $\bigcup_i \mathcal{B}(Y_i) \supset$ *vertexes of* $\mathcal{CH}(P^{'})$.

The property 1 obligates that the computed $\mathcal{B}(P^{'})$ contain the vertexes of convex hull of $P^{'}$; it is necessary that all extreme points be include as members of $\mathcal{B}(P^{'})$ in order to explore all space in which points are located, regardless of their distribution.

The property 2 states that border points of $P^{'}$ can be computed on disjoint partitions Y_i of $P^{'}$. The resulting $\mathcal{B}(P^{'})$ contain all border samples of $P^{'}$, this is because border samples are not only searched in exterior borders of the set $P^{'}$, but also within it. The size of $\bigcup_i \mathcal{B}(Y_i)$ is of course greater than the size of $\mathcal{B}(P^{'})$.

Finally the property 3 is similar to the property 1, but here the resulting non-convex hull computed on partitions Y_i of of $P^{'}$ must contain the vertexes of convex hull. If the border samples computed on partitions of $P^{'}$ contain extreme points, then not only the points in interior corners are detected but also those on the the vertexes of convex hull.

In order the detect border samples and overcome the problems of convex hull approach (interior corners and clusters of points) we propose a strategy based on three properties, if they are accomplished then those points that are not considered in the convex hull but that can be border points (according to definition in past section) can be easily detected.

3 Border Samples Detection

The novel method that we propose is based on the concave hull algorithm presented in [8], with important differences explained in the introduction of this paper, also there are some advantages over [8]: computation of border samples regardless of density distribution of points, extended to more than two dimensions and a easy concurrent implementation is possible. The method consists in three phases: 1) discretization; 2) selection of groups of adjacent boxes; 3) reduction of dimensions and computation of borders.

Firstly, a discretization of a set of points P is done by inserting each $p_i \in P$ into a binary tree $\mathcal{T}$, which represents a grid. The use of a grid helps us to avoid the explicit use of clustering algorithms to get groups of points near among them. This discretization can be seen as the mapping

$$T : P \mapsto P' \qquad (5)$$

where $P, P^{'} \in R^n$. Each leaf in $\mathcal{T}$ determine a hyper box b_i, the deeper $\mathcal{T}$ the lesser the volume of b_i. The time required to map all samples in P into the grid is $O(nlog_2(n))$. This mapping is important because it avoids more complicated and computationally expensive operations to artificially create zones of points more equally spaced, also the computation of non-convex hulls requires a set of no repeated points, if two points are repeated in P, then both are mapped to the same hyper box. All this is achieved with the mapping without requiring an additional step $O(|P|)$. During the mapping, the number of points passed trough each node of $\mathcal{T}$ is stored in a integer variable.

The second phase consists in the selection of groups of adjacent boxes in $\mathcal{T}$. The are two main intentions behind this: compute the border of a single cluster of points and control the size of it. We accomplish this two objectives by recursively traversing down $\mathcal{T}$. We stop in a node that contain less than a value of L (predefined by user) in the variable that holds the number of points that have passed through, then we recover the leaves (boxes) below the node. The set of boxes form a partition of $P^{'}$ and are refereed as Y_i. The Algorithm 1 shows the general idea of the method.

For each partition Y_i found, we first reduce the dimension and then compute its border points using algorithm shown in Algorithm 2, which works as follows. First Algorithm 2 computes $\mathcal{CH}(Y_i)$ and then each side of it is explored searching for those points that will form the non-convex hull $\mathcal{B}(P^{'})$ for the partition Y_i. The angle θ in algorithm 2 is computed using the two extreme points of each

```
Data:
$P \in R^n$: A set of points;
Result:
$\mathcal{B}(P)$: Border samples for $P$
Map P into P'                                    /* Create a binary tree $\mathcal{T}$ */
Compute partitions $Y_i$ by traversing $\mathcal{T}$                    /* Use Algorithm 1 */
Reduce dimension     /* Apply Algorithm 4, obtain $cluster_i, i = 1, 2, ...$ */
for each $cluster_i$ do
    Compute border samples for $Y_i$ within $cluster_i$              /* Algorithm 2 */
    Get back $Y_i$ to original dimension using the centroid of $cluster_i$
    $\mathcal{B}(P') \leftarrow \mathcal{B}(P') \cup$ result of previous step
end
return $\mathcal{B}(\mathcal{P})$
```

Algorithm 1. Method to compute border samples

side of the convex hull. This part of the method is crucial to compute border samples, because we are searching all points near of each side of convex hull, which are border points. These border points of each side of the convex hull are computed using the algorithm 3.

```
Data:
$Y_i$: Partition of $P$
$L$: Minimum number of candidates
Result:
$\mathcal{B}(Y_i)$: The border samples for partition $Y_i$
$\mathcal{CH} \leftarrow \mathcal{CH}(Y_i)$                  /* The sides $S = \{S_1, \dots, S_N\}$ of $\mathcal{CH}$  */
$\theta \leftarrow 0$                                               /* The initial angle */
$\mathcal{B}(Y_i) \leftarrow \varnothing$
for each side $S_i \in S$ of $\mathcal{CH}$ do
    $\mathcal{BP} \leftarrow$ Compute border points $(Y_i, S_i, L, \theta)$
    $\theta \leftarrow$ get angle $\{s_{i1}, s_{i2}\}$                             /* Update the angle */
    $\mathcal{B}(Y_i) \leftarrow \mathcal{B}(Y_i) \cup \mathcal{BP}$
end
return $\mathcal{B}(Y_i)$
```

Algorithm 2. Detection of border samples for a partition Y_i

The Algorithm 3 shows how each side of $\mathcal{CH}(Y_i)$ is explored. It is similar the presented in [8] which is based on Jarviś march but considering only local candidates, the candidates are those points located inside a box centered at point p_i being analyzed. These local points are computed quickly if Y_i have been previously sorted . The algorithm 3 always include extreme points of Y_i which produces different results from the algorithm in [8]. Also, instead of considering k-nn we use the candidates near to point p_i being analyzed (`currentPoint` in Algorithm 3).

```
Data:
$Y_i$: A partition of $P$
$S$: Side of a $\mathcal{CH}(Y_i)$
$L$: (minimum) Number of candidates; $\theta$: Previous angle.
Result:
$\mathcal{BP}$: Border points to side $S$
$firstPoint \leftarrow$ first element of $S$
$stopPoint \leftarrow$ second element of $S$
$\mathcal{BP} \leftarrow \{firstPoint\}$
$currentPoint \leftarrow firstPoint$
$previousAngle \leftarrow \theta$
while $currentPoint \neq stopPoint$ do
    if $K > |Y_i|$ then
        $L \leftarrow |Y_i|$
    end
    $candidates \leftarrow$ Get Ł elements in the box centered at $currentPoint$
    Sort $candidates$ by angle considering $previousAngle$
    $currentPoint \leftarrow$ find the first element that do not intersect $\mathcal{BP}$
    if $currentPoint$ is NOT null then
        Build a line $\mathcal{L}$ with $currentPoint$ and $stopPoint$
        if $\mathcal{L}$ intersects $\mathcal{BP}$ then
            $\mathcal{BP} \leftarrow \mathcal{BP} \cup stopPoint$
            return $\mathcal{BP}$
        end
    else
        $\mathcal{BP} \leftarrow \mathcal{BP} \cup stopPoint$
        return $\mathcal{BP}$
    end
    $\mathcal{BP} \leftarrow \mathcal{BP} \cup currentPoint$
    Remove $currentPoint$ from $X$
    $previousAngle \leftarrow$ angle between last two elements of $\mathcal{BP}$
end
return $\mathcal{BP}$
```

Algorithm 3. Computation of border points for S_i

For higher than two dimensions we create partitions on them to temporally reduce the dimension of the $P \in R^n$ in several steps. For each dimension we create one dimensional clusters, the number of cluster corresponds to the partitions of the dimension being reduced, then we fixed the value of that partition to be the center of the corresponding cluster. This process is repeated on each dimension. The final bi-dimensional subsets used are formed by considering in decreasing order with respect to the number of partitions of each dimension. We compute border samples and then get them back to their original dimension taking the previously fixed values.

In order to quickly compute the clusters on each feature of data set, we use a similar algorithm to that presented in [12]. The basic idea of the on-line dimensional clustering as follows: if the distance from a sample to the center of a group is less than a previously defined distance L, then the sample belongs to this group. When new data are obtained, the center and the group should also change. The Euclidean distance at time k is defined by eq. (6)

$$d_{k,x} = \left(\sum_{i=1}^{n} \left[\frac{x_i(k) - \overline{x}^j}{x_{imax} - x_{imin}} \right]^2 \right)^{\frac{1}{2}} \tag{6}$$

Where n is the dimension of sample x, $\overline{x}^j$ is the center of the j^{th} cluster, $x_{imax} = max_k \left\{ x_i(k) \right\}$ and $x_{imin} = min_k \left\{ x_i(k) \right\}$.

The center of each cluster can be recursively computed using (7):

$$\overline{x}_i^j{}_{k+1} = \frac{k-1}{k} \overline{x}_i^j{}_k + \frac{1}{k} x_i(k) \tag{7}$$

The Algorithm 4 shows how to compute the partition of one dimension of a given training data set. This algorithm 4 is applied in each dimension, and produces results in linear time with the size of the training data set.

```
Data:
$\mathcal{X}_m$: The values of the $m^{th}$ feature of training data set $\mathcal{X}$
Result:
$\mathcal{C}_j$: A number of one dimensional clusters(partitions) of $m^{th}$ feature of $\mathcal{X}$ and its
corresponding centers.
$\mathcal{C}_1 = x(1)$                /* First cluster is the first arrived sample. */
$\overline{x}^1 = x(1)$
for each received data $x(k)$ do
    Use eq. (6) to compute distance $d_{k,x}$ from $x(k)$ to cluster $\mathcal{C}_j$
    if $d_{k,x} \leq L$ then
        $x(k)$ is kept in cluster $j$
        Update center using eq. (7).
    else
        $x(k)$ belongs to a new cluster $\mathcal{C}_{j+1}$, i.e., $\mathcal{C}_{j+1} = x(k)$
        $\overline{x}^{j+1} = x(k)$
    end
end
        /* If the distance between two groups centers is more than the
required distance $L$ */
if $\sum_{i=1}^{n} \left[ \overline{x}^p - \overline{x}_i^q \right]^2 \leq L$ then
    The two clusters ($\mathcal{C}_p$ and $\mathcal{C}_q$) are combined into one group, the center of the
    new group may be any of the two centers.
end
return $SV_1 \bigcup SV_2$
```

Algorithm 4. Feature partition algorithm

4 Application on a Data Mining Task

In order to show the effectiveness of the proposed method, we apply the developed algorithms on several data sets and then train a SVM using the detected border points.

All experiments were run on a computer with the following features: Core 2 Duo 1.66 GHz processor, 2.5 GB RAM, linux Fedora 15 operating system installed. The algorithms were implemented in the Java language. The maximum amount of random access memory given to the java virtual machine was set to 1.6 GB for each one of the runs.

For all data sets the training data set was built by randomly choosing the 70% of the whole data set read from disk, the rest of samples were used as testing data set.

The data sets are stored as plain text files in the attribute relation file format. The time used to read the data sets from hard disk was not taken into account for the reported results of all the experiments, as usual in literature, i.e., the measurements were taken from the time when a data set was loaded into memory to the time when the model has been calibrated, i.e., the reported times correspond to the computation of border samples and the training of SVM. The reported results are the average of 10 runs of each experiment.

In order to compare the performance of the proposed algorithm two SVMs are trained using LibSVM library. The first SVM is trained with the entire data set whereas the second SVM is trained using only the border samples recovered using the proposed method. In both cases the corresponding training times and achieved accuracy are measured and compared. The kernel used in all experiments is a radial basis function.

Experiment 1. In this experiment, we use a data set similar to the checkerboard one [13]. Table 1 shows a resume of the data set. The difference with the original is that the data set used in the experiment contains 50000 samples grouped in a similar distribution as shown in figure 4. The squares can overlap in no more than 10%. Note that the number of samples have kept small to clarify the view.

Table 1. Data set Checkerboard2 used in experiment 1

Data set	Features	Size ($y_i = +1/y_i = -1$)
Checkerboard2	2	25000 (12500/12500)

The Checkerboard2 is a linearly inseparable data set. The RBF kernel was used with the parameter $\gamma = 0.055$. Table 2 shows the results of the Experiment1. Column T_{br} in the table refers to the **time for the computation of border samples**, whereas T_{tr} is **the training time**, both are in milliseconds. The column *Time* **is the time elapsed from the load of data set in memory to the time when training of SVM is done**, also it is measured in milliseconds. The column #SV is the number of support vectors and #BS is the number of border samples recovered by the proposed algorithm. The first row of results shows the results using border samples detected with the proposed algorithm whereas the second one is for LibSVM using the entire data set.

Table 2. Results for Checkerboard2 like data set (25000 samples)

T_{bs}	T_{tr}	Time	#SV	#BS	Acc	Training data set
1618	4669	**6287**	2336	2924	**89.9**	*Only Border Samples*
		27943	4931		**90.3**	Whole data set

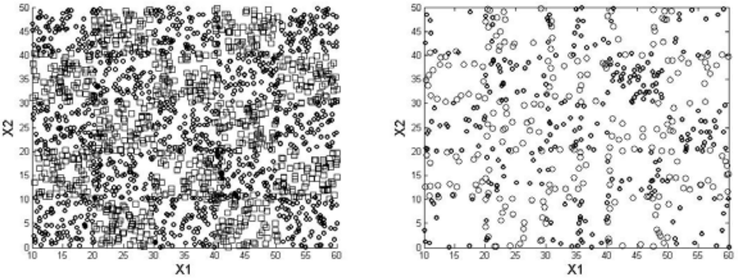


Fig. 3. Example of Checkerboard data set and border samples computed with the proposed algorithm

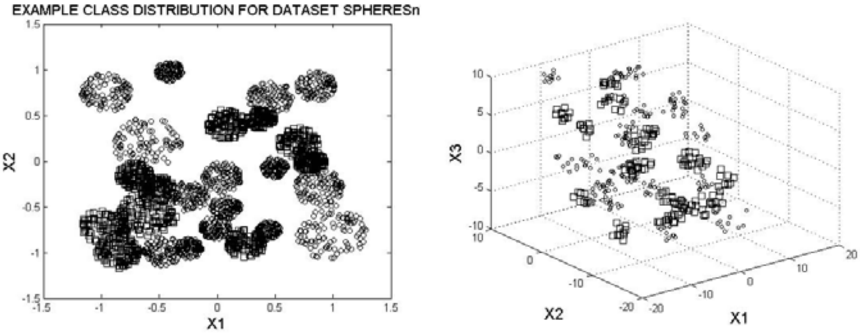


Fig. 4. Example of the class Distribution for data set **Spheres***2* and **Spheres***3*. In higher dimensions a similar random distribution occurs. Circle: $y_i = +1$, Square: $y_i = -1$.

Fig. 4 can be appreciated the border samples detected from data set Checkerboard. The method successfully compute border samples and produces a reduced version of Checkerboard, containing only border points. This samples are used to train SVM, which accelerated the training time as can be seen in table 2.

Experiment 2. In the second experiment, we use a data set of size up to 250000 samples and the number of dimensions is increased up to 4.. The data set is synthetic, composed of dense hyper dimensional balls with random radius and centres. The synthetic data set Spheres$n$ consists in a a number of hyper spheres whose center is randomly located in a n dimensional space. Each sphere has a radius of random length and contains samples having the same label. The hyper spheres can overlap in no more than 10% of the greater radius. Fig. 4

shows an example of data set Spheres$n$ for n=2 and n=3. Again the number of samples have kept small to clarify the view. Similar behaviour occurs in higher dimensions. In the Table 3 can be seen the number of samples and the dimension of data set used in the experiment 2.

Table 3. Data set Spheres$n$ used in experiment 2

Data set	Features	Size ($y_i = +1/y_i = -1$)
Spheres2	2	50000 (16000/34000)
Spheres4	4	200000 (96000/104000)

The training and testing data sets were built by randomly choosing 70% and 30% respectively from the whole data set. For all runs in experiment 2, the parameter $\gamma = 0.07$.

Table 4. Results for Spheres*2* data set (50000 samples)

T_{br}	T_{tr}	Time	#SV	#BS	Acc	Training data set
2635	2887	**5522**	626	2924	**98.4**	*Only Border Samples*
		69009	1495		**98.6**	Whole data set

Table 5. Results for Spheres*4* data set ((200000 samples))

T_{br}	T_{tr}	Time	#SV	#BS	Acc	Training data set
6719	2001	**8720**	627	4632	**98.3**	*Only Border Samples*
		53643	1173		**99.8**	whole data set

Results show that accuracy of the classifier trained using only border samples is slightly degraded but the training times of SVM are reduced considerably. Which agree with the fact that border samples were successfully recognized from training data set.

5 Conclusions

We proposed a method to compute the border samples of a set of points in a multidimensional space. The results of experiments show that the effectiveness of the method on classification task using SVM, the algorithms can quickly obtain border samples that are used to train SVM yielding similar accuracy to the obtained using the whole data set but with the advantage of consuming considerably less time. We are currently working on an incremental version of the algorithm to compute border samples.

References

1. Edelsbrunner, H., Mücke, E.P.: Three-dimensional alpha shapes. ACM Trans. Graph. 13(1), 43–72 (1994)
2. Bader, M.A., Sablatnig, M., Simo, R., Benet, J., Novak, G., Blanes, G.: Embedded real-time ball detection unit for the yabiro biped robot. In: 2006 International Workshop on Intelligent Solutions in Embedded Systems (June 2006)
3. Zhang, J., Kasturi, R.: Weighted boundary points for shape analysis. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 1598–1601 (August 2010)
4. Hoogs, A., Collins, R.: Object boundary detection in images using a semantic ontology. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006, p. 111 (June 2006)
5. Edelsbrunner, H., Kirkpatrick, D., Seidel, R.: On the shape of a set of points in the plane. IEEE Transactions on Information Theory 29(4), 551–559 (1983)
6. Galton, A., Duckham, M.: What is the Region Occupied by a Set of Points? In: Raubal, M., Miller, H.J., Frank, A.U., Goodchild, M.F. (eds.) GIScience 2006. LNCS, vol. 4197, pp. 81–98. Springer, Heidelberg (2006)
7. Xia, C., Hsu, W., Lee, M., Ooi, B.: Border: efficient computation of boundary points. IEEE Transactions on Knowledge and Data Engineering 18(3), 289–303 (2006)
8. Moreira, J.C.A., Santos, M.Y.: Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. In: GRAPP (GM/R), pp. 61–68 (2007), http://dblp.uni-trier.de
9. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Heidelberg (2008)
10. O'Rourke, J.: Computational Geometry in C. Cambridge University Press (1998), hardback ISBN: 0521640105; Paperback: ISBN 0521649765, http://maven.smith.edu/~orourke/books/compgeom.html
11. Noble, B., Daniel, J.W.: Applied Linear Algebra, 3rd edn. (1988)
12. Yu, W., Li, X.: On-line fuzzy modeling via clustering and support vector machines. Information Sciences 178, 4264–4279 (2008)
13. Ho, T., Kleinberg, E.: Checkerboard data set (1996), http://www.cs.wisc.edu/math-prog/mpml.html

An Active System for Dynamic Vertical Partitioning of Relational Databases

Lisbeth Rodríguez, Xiaoou Li, and Pedro Mejía-Alvarez

Department of Computer Science, CINVESTAV-IPN, Mexico D.F., Mexico
lisbethr@computacion.cs.cinvestav.mx,
{lixo,pmalvarez}@cs.cinvestav.mx

Abstract. Vertical partitioning is a well known technique to improve query response time in relational databases. This consists in dividing a table into a set of fragments of attributes according to the queries run against the table. In dynamic systems the queries tend to change with time, so it is needed a dynamic vertical partitioning technique which adapts the fragments according to the changes in query patterns in order to avoid long query response time. In this paper, we propose an active system for dynamic vertical partitioning of relational databases, called DYVEP (DYnamic VErtical Partitioning). DYVEP uses active rules to vertically fragment and refragment a database without intervention of a database administrator (DBA), maintaining an acceptable query response time even when the query patterns in the database suffer changes. Experiments with the TPC-H benchmark demonstrate efficient query response time.

Keywords: Active systems, active rules, dynamic vertical partitioning, relational databases.

1 Introduction

Vertical partitioning has been widely studied in relational databases to improve query response time [1-3]. In vertical partitioning, a table is divided into a set of fragments, each with a subset of attributes of the original table and defined by a vertical partitioning scheme (VPS). Fragments consist of smaller records, therefore, fewer pages from secondary memory are accessed to process queries that retrieve or update only some attributes from the table, instead of the entire record [3].

Vertical partitioning can be static or dynamic [4]. Most works consider a static vertical partitioning based on a priori probabilities of queries accessing database attributes in addition to their frequencies which are available during the analysis stage. It is more effective for a database to dynamically check the goodness of a VPS to determine whenever refragmentation is necessary [5].

Static vertical partitioning works only consider that the queries that operate on the relational database are static and a VPS is optimized for such queries. Nevertheless, applications like multimedia, e-business, decision support, and geographic information systems are accessed by many users simultaneously. Therefore, queries

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 273–284, 2011.

tend to change over time, and a refragmentation of the database is needed when query patterns and database scheme have undergone sufficient changes.

Dynamic vertical partitioning techniques automatically trigger the refragmentation process if it is determined that the VPS in place has become inadequate due to a change in query patterns or database scheme. This implies to develop a system which can trigger itself and make decision on their own.

Active systems are able to respond automatically to events that are taking place either inside or outside the system itself. The central part of those systems is a set of active rules which codifies the knowledge of domain experts [6]. Active rules constantly monitor systems and user activities. When an interesting event happens, they respond by executing certain procedures related either to the system or to the environment [7].

The general form of an active rule is the following:

ON *event*
IF *condition*
THEN *action*

An event is something that occurs at a point in time, e.g., a query in database operation. The condition examines the context in which the event has taken place. The action describes the task to be carried out by the rule if the condition is fulfilled once an event has taken place. Several applications, such as smart homes, sensor and active databases integrate active rules for the management of some of their important activities [8].

In this paper, we propose an active system for dynamic vertical partitioning of relational databases, called DYVEP (DYnamic VErtical Partitioning). Active rules allow DYVEP to automatically monitor the database in order to collect statistics about queries, detect changes in query patterns, evaluate the changes and when the changes are greater than a threshold, trigger the refragmentation process.

The rest of the paper is organized as follows: in Section 2 we give an introduction on dynamic vertical partitioning. In Section 3 we present the architecture of DYVEP. Section 4 presents the implementation of DYVEP, and finally Section 5 is our conclusion.

2 Dynamic Vertical Partitioning

2.1 Motivation

Vertical partitioning can be static and dynamic [5]: In the former, attributes are assigned to a fragment only once at creation time, and then their locations are never changed. This approach has the following problems:

1. The DBA has to observe the system for a significant amount of time until probabilities of queries accessing database attributes in addition to their frequencies are discovered before the partitioning operation can take place. This is called an analysis stage.

2. Even then, after the partitioning process is completed, nothing guarantees that the real trends in queries and data have been discovered. Thus the VPS may not be good. In this case, the database users may experience very long query response time [14].
3. In some dynamic applications, queries tend to change over time and a VPS is implemented to optimize the response time for one particular set of queries. Thus, if the queries or their relative frequencies change, the partitioning result may no longer be adequate.
4. With static vertical partitioning methods, refragmentation is a heavy task and only can be performed manually when the system is idle [11].

In contrast, with dynamic vertical partitioning, attributes are being relocated if it is determined that the VPS in place has become inadequate due to a change in query information. We develop DYVEP to improve the performance of relational database systems. Using active rules, DYVEP can monitor queries run against the database in order to accumulate the accurate information to perform the vertical partitioning process, eliminating the cost of the analysis stage. It also automatically reorganizes the fragments according to the changes in query patterns and database scheme, achieving good query performance at all times.

2.2 Related Work

Liu Z. [4] presents an approach for dynamic vertical partitioning to improve query performance in relational databases, this approach is based on the feedback loop used in automatic performance tuning, which consists of observation, prediction and reaction. It observes the change of workload to detect a relatively low workload time, and then it predicts the coming workload based on the characteristics of current workload and implements the new vertical partitions.

Reference [9] integrates both horizontal and vertical partitioning into automated physical database design. The main disadvantage of this work is that they only recommend the creation of vertical fragments but the DBA has to create the fragments. DYVEP has a partitioning reorganizer which creates automatically the fragments on disk.

Autopart [10] is an automated tool that partitions the relations in the original database according to a representative workload. Autopart receives as input a representative workload and designs a new schema using data partitioning, one drawback of this tool is that the DBA has to give the workload to autopart. In contrast, DYVEP collects the SQL statements when they are executed.

Dynamic vertical partitioning is also called dynamic attribute clustering. Guinepain and Gruenwald [1] present an efficient technique for attribute clustering that dynamically and automatically generates attribute clusters based on closed item sets mined from the attributes sets found in the queries running against the database.

Most dynamic clustering techniques [11-13] consist of the following modules: a statistic collector (SC) that accumulates information about the queries run and data returned. The SC is in charge of collecting, filtering, and analyzing the statistics. It is responsible for triggering the Cluster Analyzer (CA). The CA determines the best possible clustering given the statistics collected. If the new clustering is better than the one in place, then CA triggers the reorganizer that physically reorganizes the data on disk [14]. The database must be monitored to determine when to trigger the CA and the reorganizer.

To the best of our knowledge there are not works related to dynamic vertical partitioning using active rules. Dynamic vertical partitioning can be effectively implemented as an active system because active rules are expressive enough to allow specification of a large class of monitoring tasks and they do not have noticeable impact on performance, particularly when the system is under heavy load. Active rules are amenable to implementation with low CPU and memory overheads [15].

3 Architecture of DYVEP

In order to get good query performance at any time, we propose DYVEP, which is an active system for dynamic vertical partitioning of relational databases. DYVEP monitors queries in order to accumulate relevant statistics for the vertical partitioning process, it analyzes the statistics in order to determine if a new partitioning is necessary, in such case; it triggers the Vertical Partitioning Algorithm (VPA). If the VPS is better that the one in place, then the system reorganizes the scheme. Using active rules, DYVEP can react to the events generated by users or processes, evaluate conditions and if the conditions are true, then execute the actions or procedures defined.

The architecture of DYVEP is shown in Fig. 1. DYVEP is composed of 3 modules: *Statistic Collector*, *Partitioning Processor*, and *Partitioning Reorganizer*.

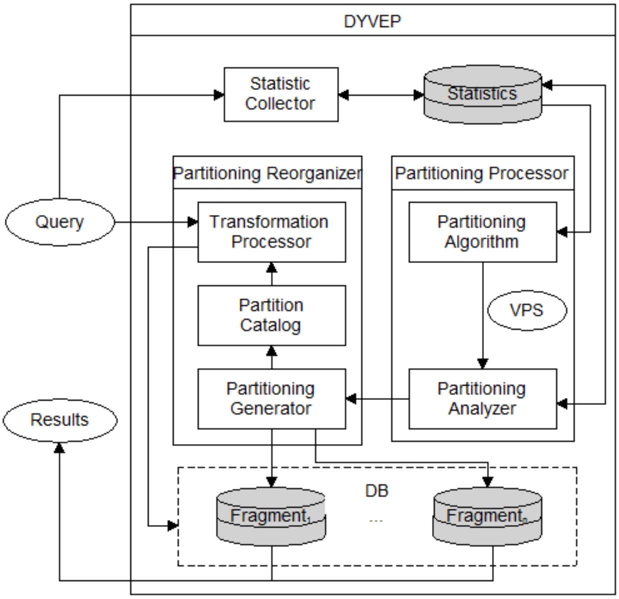


Fig. 1. Architecture of DYVEP

3.1 Statistic Collector

The statistic collector accumulates information about the queries (such as id, description, attributes used, access frequency) and the attributes (name, size). When DYVEP is executed for first time in the database, the statistic collector creates the tables queries (QT), attribute_usage_table (AUT), attributes (AT) and statistics (stat) and a set of active rules in such tables.

After initialization, when a query (q_i) is run against the database, the statistic collector verifies if the query is not stored in QT; in that case it assigns an id to the query, stores its description, and sets its frequency to 1 in QT. If the query is already stored in QT, only its frequency is increased by 1. This is defined by the following active rule:

Rule 1
ON $q_i \in Q$
IF $q_i \notin QT$
THEN insert QT (*id, query, freq*) values (*id_ q_i, query_ q_i, 1*)
ELSE update QT set *freq*=old.freq+1 where *id*=id_ q_i

In order to know if the query is already stored in QT, the statistic collector has to analyze the queries. Two queries are considered equal if they use the same attributes, for example if we have the queries:

q_1: SELECT *A, B* FROM T
q_2: SELECT SUM (*B*) FROM T WHERE *A*=Value

If q_1 is already stored in QT and q_2 is run against the database, the statistic collector analyzes q_2 in order to know the attributes used by the query, and compares q_2 with the queries already stored in QT, since q_1 uses the same attributes then its frequency is increased by 1.

The statistic collector also registers the changes in the information of queries and attributes over time and compares the current changes (*currentChange*) with the previous changes (*previousChange*) in order to determine if they are enough to trigger the VPA. For example, when a query is inserted or deleted in QT after initialization, the changes in queries are calculated. If the changes are greater than a threshold, then VPA is triggered.

The changes in queries are calculated as the number of inserted or deleted queries after a refragmentation divided by the total number of queries before refragmentation. For example, if QT had 8 queries before the last refragmentation and one query is inserted after refragmentation, then the change in queries is equal to 1/8*100=12.5%. If the value of the threshold is 10%, then VPA will be triggered.

The threshold is updated after each refragmentation and it is defined as previousChange plus *currentChange* divided by two.

The following rules are implemented in the statistic collector:

Rule 2
ON insert or delete QT
THEN update stat set *currentNQ*=*currentNQ*+1

Rule 3
ON update stat.*currentNQ*
IF *currentNQ*>0 and *previousNQ*>0
THEN update stat set *currentChange=currentNQ/previousNQ**100

Rule 4
ON update stat.*currentChange*
IF *currentChange*>threshold
THEN call VPA

3.2 Partitioning Processor

The partitioning processor has two components: the partitioning algorithm and the partitioning analyzer. The partitioning algorithm determines the best VPS given the collected statistics, which is presented in Algorithm 1.

The partitioning analyzer detects if the new VPS is better than the one in place, then the partitioning analyzer triggers the partitioning generator in the partitioning reorganizer module. This is defined using an active rule:

Rule 5
ON *new VPS*
IF *new_VPS_cost<old_VPS_cost*
THEN call *partitioning_generator*

Algorithm 1. Vertical Partitioning Algorithm

```
input: QT: Query Table
output: Optimal vertical partitioning scheme (VPS)
begin
{Step 1: Generating AUT}
   getAUT(QT, AUT)
   {generate the AUT from QT}
{Step 2: Getting the optimal VPS}
   getVPS(AUT, VPS)
   {get the optimal VPS using the AUT of step 1}
end. {VPA}
```

3.3 Partitioning Reorganizer

The partitioning reorganizer physically reorganizes the fragments on disk. It has three components: a partitioning generator, a partition catalog and a transformation processor. The partitioning generator creates the new VPS, deletes the old scheme and registers the changes in the partitioning catalog. The partitioning catalog contains the location of the fragments and the attributes of each fragment. The transformation processor transforms the queries so that they can execute correctly in the partitioned domain. This transformation involves replacing attribute accesses in the original query definition with appropriate path expressions. The transformation processor uses the partitioning catalog to determine the new attribute location.

When a query is submitted to the database DYVEP triggers the transformation processor, which changes the definition of the query according to the information located in the partitioning catalog. The transformation processor sends the new query to the database; the database then executes the query and provides the results.

4 Implementation

We have implemented DYVEP using triggers inside the open source PostgreSQL object-relational database system running on a single processor 2.67-GHz Intel (R) Core(TM) i7CPU with 4 GB of main memory and 698-GB hard drive.

4.1 Benchmark

As an example, we use the TPC-H benchmark [16], which is an ad-hoc, decision support benchmark widely used today in evaluating the performance of relational database systems. We use the partsupp table of TPC-H 1 GB; partsupp has 800,000 tuples and 5 attributes.

In most of today's commercial database systems, there is not native DDL support for defining vertical partitions of a table [9]. Therefore, it can be implemented as a relational table, a relational view, an index or a materialized view. If the partition is implemented as a relational table, it may cause a problem of optimal choice of partition for a query. For example, suppose we have table

partsupp
(*ps_partkey,*
ps_suppkey,
ps_availqty,
ps_supplycost,
ps_comment),

Partitions of partsupp::

partsupp_1(*ps_partkey, ps_psavailqty, ps_suppkey, ps_supplycost*)
partsupp_2(*ps_partkey, ps_comment*)
Where *ps_partkey* is the primary key. Considering a query:

SELECT *ps_partkey*, *ps_comment* FROM partsupp

The query of selection of partsupp cannot be transformed to selection from partsupp_2 by query optimizer automatically. If the partition is implemented as a materialized view, the query processor in the database management system can detect the optimal materialized view for a query and be able to rewrite the query to access the optimal materialized view. If the partitions are implemented as indexes over the relational tables, the query processor is able to detect that horizontal traversal of an index is equivalent to a full scan of a partition. Therefore implementing the partitions either as a materialized view or index allows the changes of the partition as transparent to the applications [4].

4.2 Illustration

DYVEP is implemented as an SQL script, the DBA who wants to partition a table executes only once DYVEP.sql in the database which contains the table to be partitioned. DYVEP will detect that it is the first execution and will create the tables, functions and triggers to implement the dynamic vertical partitioning.

Step 1. The first step of DYVEP is to create an initial vertical partitioning, to generate this, the Statistic collector of DYVEP analyzes the queries stored in the statement log and copies the queries run against the table to be partitioned in the table queries (QT). To implement the **Rule 1** on this table, we create a trigger called *insert_queries*.

Step 2. When all the queries has been copied for the statistic collector, then it triggers the vertical partitioning algorithm, DYVEP can use any algorithm that uses as input the attribute_usage_table (AUT), as an example, the vertical partitioning algorithm implemented in DYVEP is the Navathe's algorithm [2], we selected this algorithm because is a classical vertical partitioning algorithm.

Step 3. The partitioning algorithm first will get the AUT from the QT, the AUT has two triggers for each attribute of the table to be fragmented, one trigger for insert and delete and one for update, in this case we have the triggers *inde_ps_partkey*, *update_ps_partkey*, etc., these triggers provide the ability to update the attribute_affinity_table (AAT) when the frequency or the attributes used by the query suffer changes in the AUT, an example of rule definition for the attribute *ps_partkey* is

Rule 6
ON update AUT
IF new.*ps_partkey*=true
THEN update AAT set *ps_partkey*=*ps_partkey*+new.*frequency* where attribute=*ps_partkey*

Step 4. When the AAT is updated, a procedure called BEA is triggered, a rule definition for this is:

Rule 7
ON update AAT
THEN call BEA

BEA is the Bond Energy Algorithm [17], which is a general procedure for permuting rows and columns of a square matrix in order to obtain a semiblock diagonal form. The algorithm is typically applied to partition a set of interacting variables into subsets which interact minimally. The application of the procedure BEA to the AAT generates the clustered affinity table (CAT),

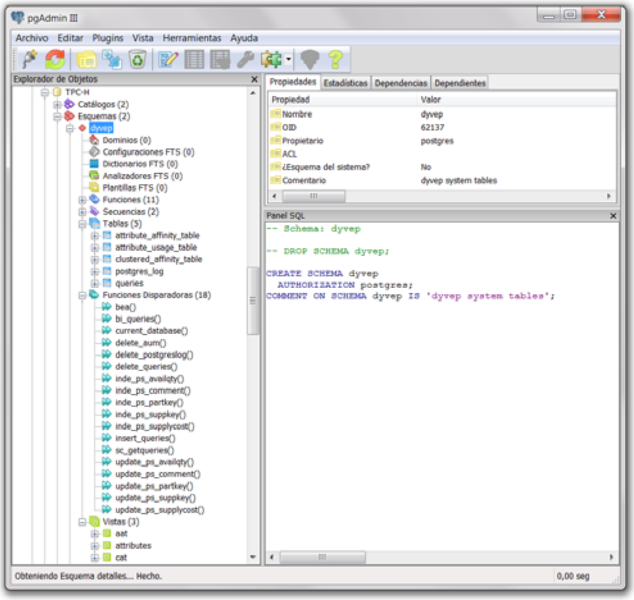
Step 5. Once CAT has been generated, a procedure called partition is triggered which receives as input the CAT and gets the vertical partitioning scheme (VPS).

Step 6. When the initial VPS is obtained, the partitioning algorithm triggers the partitioning generator which materializes the VPS, i.e., creates the fragments on disk. The active rule for this is:

Rule 8
ON NEW VPS
IF *VPS_status*=initial
THEN call *partitioning_generator*

Step 7. The partitioning generator implements the fragments as materialized views, so the query processor of PostgreSQL can detect the optimized materialized view for a query and is able to rewrite the query to access the optimal materialized view instead of the complete table. This provides fragmentation transparency to the database.

A screenshot of DYVEP is given in Fig. 2. A scheme called DYVEP is created in the database. In such scheme, all the tables (queries, attribute_usage_table, attribute_affinity_table, clustered_affinity_table) from the DYVEP system are located, the triggers *inde_attributename*, *update_attributename* are generated automatically by DYVEP according to the view attributes, therefore the number of triggers in our system will depend on the number of attributes of the table to fragment.



Propiedad	Valor
Nombre	dyvep
OID	62137
Propietario	postgres
ACL	
¿Esquema del sistema?	No
Comentario	dyvep system tables

```
-- Schema: dyvep

-- DROP SCHEMA dyvep;

CREATE SCHEMA dyvep
  AUTHORIZATION postgres;
COMMENT ON SCHEMA dyvep IS 'dyvep system tables';
```

Obteniendo Esquema detalles... Hecho. 0,00 seg

Fig. 2. Screenshot of DYVEP in PostgreSQL

4.3 Comparisons

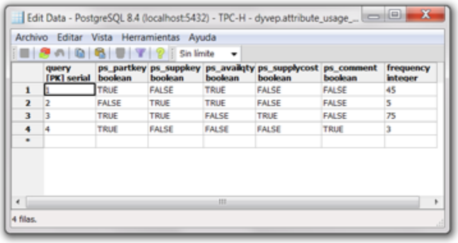
Having the following queries

q_1: SELECT SUM(*ps_availqty*) FROM partsupp WHERE *ps_partkey*=Value
q_2: SELECT *ps_suppkey*, *ps_availqty* FROM partsupp

q_3: SELECT *ps_suppkey*, *ps_supplycost* FROM partsupp WHERE *ps_partkey*=Value
q_4: SELECT *ps_comment*, *ps_partkey* FROM partsupp

DYVEP got the attribute usage table of Fig. 3. The VPS obtained by DYVEP according to the attribute usage table was

partsupp_1 (ps_*partkey*, *ps_psavailqty*, *ps_suppkey*, *ps_supplycost*)
partsupp_2 (*ps_partkey*, *ps_comment*)


Edit Data - PostgreSQL 8.4 (localhost:5432) - TPC-H - dyvep.attribute_usage_...
Archivo Editar Vista Herramientas Ayuda
Sin límite

	query [PK] serial	ps_partkey boolean	ps_suppkey boolean	ps_availqty boolean	ps_supplycost boolean	ps_comment boolean	frequency integer
1	1	TRUE	FALSE	TRUE	FALSE	FALSE	45
2	2	FALSE	TRUE	TRUE	FALSE	FALSE	5
3	3	TRUE	TRUE	FALSE	TRUE	FALSE	75
4	4	TRUE	FALSE	FALSE	FALSE	TRUE	3
*							

4 filas.

Fig. 3. Attribute Usage Table

In Table 1 we can see the execution time of these queries in TPC-H not partitioned (NP) vs. vertically partitioned using DYVEP. As we can see, the execution time of the queries in TPC-H vertically partitioned using DYVEP is lower than in a TPC-H not partitioned, therefore DYVEP can generate schemes that can significantly improve query execution, even without the use of any indexes.

Table 1. Comparison of query execution time

TPC_H	q_1	q_2	q_3	q_4
NP	47 ms	16770 ms	38 ms	108623 ms
DYVEP	15 ms	16208 ms	16 ms	105623 ms

5 Conclusion and Future Work

A system architecture for performing dynamic vertical partitioning of relational databases has been designed, which can adaptively modify the VPS of a relational database using active rules within efficient query response time. The main advantages of DYVEP over other approaches are:

1. Static vertical partitioning strategies [2] take into account an a priori analysis stage of the database in order to collect the necessary information to perform the vertical partitioning process, also in some automated vertical partitioning tools [9, 10] it is necessary that the DBA gives as input the workload. In contrast, DYVEP implements an active-rule based statistic collector which accumulates

information about attributes, queries and fragments without the explicit intervention of the DBA.

2. When the information of the queries changes in the static vertical partitioning strategies, then the fragment configuration will remain in the same way and will not implement the best solution. In DYVEP the fragment configuration will change dynamically according to the changes in the information of the queries in order to find the best solution and not affect the performance of the database.
3. The vertical partitioning process in the static approaches is performed outside of the database and when the solution is found the vertical fragments are materialized. In DYVEP all the vertical partitioning process is implemented inside the database using rules, the attribute usage matrix (AUM) used by most of the vertical partitioning algorithms is implemented as a database table (AUT) in order to use rules to change the fragment configuration automatically.
4. Some automated vertical partitioning tools only recommend the optimal vertical partitioning configuration but they leave the creation of the fragments to the DBA [9], DYVEP has an active rule-based partitioning reorganizer that automatically creates the fragments on disk when is triggered by the partitioning analyzer.

In the future, we want to extend our results to multimedia database system. Multimedia database systems are highly dynamic, so the advantages of DYVEP would be seen much clearly, especially on reducing the query response time.

References

1. Guinepain, S., Gruenwald, L.: Using Cluster Computing to support Automatic and Dynamic Database Clustering. In: Third International Workshop on Automatic Performance Tuning (IWAPT), pp. 394–401 (2008)
2. Navathe, S., Ceri, S., Wiederhold, G., Dou, J.: Vertical Partitioning Algorithms for Database Design. ACM Trans. Database Syst. 9(4), 680–710 (1984)
3. Guinepain, S., Gruenwald, L.: Automatic Database Clustering Using Data Mining. In: 17th Int. Conf. on Database and Expert Systems Applications, DEXA 2006 (2006)
4. Liu, Z.: Adaptive Reorganization of Database Structures through Dynamic Vertical Partitioning of Relational Table., MCompSc thesis, School of Information Technology and Computer Science, University of Wollongong (2007)
5. Sleit, A., AlMobaideen, W., Al-Areqi, S., Yahya, A.: A Dynamic Object Fragmentation and Replication Algorithm in Distributed Database Systems. American Journal of Applied Sciences 4(8), 613–618 (2007)
6. Chavarría-Baéz, L., Li, X.: Structural Error Verification in Active Rule Based-Systems using Petri Nets. In: Gelbukh, A., Reyes-García, C.A. (eds.) Fifth Mexican International Conference on Artificial Intelligence (MICAI 2006), pp. 12–21. IEEE Computer Science (2006)
7. Chavarría-Baéz, L., Li, X.: ECAPNVer: A Software Tool to Verify Active Rule Bases. In: 22nd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 138–141 (2010)

8. Chavarría-Baéz, L., Li, X.: Termination Analysis of Active Rules - A Petri Net Based Approach. In: IEEE International Conference on Systems, Man and Cybernetics, San Antonio, Texas, USA, pp. 2205–2210 (2009)
9. Agrawal, S., Narasayya, V., Yang, B.: Integrating Vertical and Horizontal Partitioning into Automated Physical Database Design. In: Proc. of the 2004 ACM SIGMOD Int. Conf. on Management of Data, pp. 359–370 (2004)
10. Papadomanolakis, E., Ailamaki, A.: AutoPart: Automating Schema Design for Large Scientific Databases Using Data Partitioning. CMU Technical Report, CMU-CS-03-159 (2003)
11. Darmont, J., Fromantin, C., Régnier, S., Gruenwald, L., Schneider, M.: Dynamic Clustering in Object-Oriented Databases: An Advocacy for Simplicity. In: Dittrich, K.R., Oliva, M., Rodriguez, M.E. (eds.) ECOOP-WS 2000. LNCS, vol. 1944, pp. 71–85. Springer, Heidelberg (2001)
12. Gay, J.Y., Gruenwald, L.: A Clustering Technique for Object Oriented Databases. In: Tjoa, A.M. (ed.) DEXA 1997. LNCS, vol. 1308, pp. 81–90. Springer, Heidelberg (1997)
13. McIver Jr., W.J., King, R.: Self-Adaptive, on-Line Reclustering of Complex Object Data. In: Proc. of the 1994 ACM SIGMOD Int. Conf. on Management of Data (1994)
14. Guinepain, S., Gruenwald, L.: Research Issues in Automatic Database Clustering. SIGMOD Record 34(1), 33–38 (2005)
15. Chaudhuri, S., Konig, A.C., Narasayya, V.: SQLCM: a Continuous Monitoring Framework for Relational Database Engines. In: Proc. of the 20th Int. Conf. on Data Engineering, ICDE (2004)
16. Transaction Processing Performance Council TPC-H benchmark, `http://www.tpc.org/tpch`
17. McCormick, W.T., Schweitzer, P.J., White, T.W.: Problem Decomposition and Data Reorganization by a Clustering Technique. Operations Research 20(5), 973–1009 (1972)

Efficiency Analysis in Content Based Image Retrieval Using RDF Annotations

Carlos Alvez[1] and Aldo Vecchietti[2]

[1] Facultad de Ciencias de la Administración, Universidad Nacional de Entre Ríos
Concordia, 3200, Argentina
[2] INGAR – UTN, Facultad Regional Santa Fe
Santa Fe, S3002GJC, Argentina

Abstract. Nowadays it is common to combine low-level and semantic data for image retrieval. The images are stored in databases and computer graphics algorithms are employed to get the pictures. Most of the works consider both aspects separately. In this work, using the capabilities of a commercial ORDBMS a reference architecture was implemented for recovering images, and then a performance analysis is realized using several index types to search some specific semantic data stored in the database via RDF triples. The experiments analyzed the mean recovery time of triples in tables having a hundred of thousands to millions of triples. The performance obtained using Bitmap, B-Tree and Hash Partitioned indexes are analyzed. The results obtained with the experiences performed are implemented in the reference architecture in order to speed up the pattern search.

Keywords: Image retrieval, Semantic data, RDF triples, Object-Relational Database.

1 Introduction

Recovering images by content in a database requires the use of metadata which can be of several types: low-level describing physical properties like color, texture, shape, etc.; or high level metadata describing the image: the people on it, the geographic place or the action pictured, e.g. a car race.

Most of the works dealing with image recovering are limited by the difference between the low-level information and the high level semantic annotations. This difference is due to the diverse perception between the low-level data extracted by the programs, and the interpretation the user has for the image [1]. To cover this limitations the actual tendency is to combine in the same approach the low-level and semantic data. On other hand most of the articles in the open literature treat separately the database management aspects of the image retrieval from the computer vision issues [2]; however, in the commercial nowadays Data Base Management Systems (DBMS) it is possible to get sophisticated tools to handle and process high level data, having the capacity to formulate ontology assisted queries and/or semantic inferences.

In this sense, Alvez and Vecchietti [3] presented a software architecture to recover images from an Object Relational Database Management System (ORDBMS) [4]

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 285–296, 2011.

using physical and semantic information. This architecture behaves as an extension of the SQL language in order to facilitate the usability of the approach. The low-level and high level information are combined maximizing the use of the tools provided by the DBMS. The architecture is based on several User Defined Types (UDT) containing attributes and methods needed to recover images based on both data types. The semantic information is added by means of the RDF (Resource Description Framework) language and RDF Schema. In this work it was shown that, although the RDF language was created for data representation in the Word Wide Web, it can be perfectly used to recover images in the database. The main advantages of using RDF/RDFS are its simplicity and flexibility, since by means of a triple of the form (*subject property object*) it is possible to represent a complete reference ontology or classes and concepts of that ontology and to make inferences among the instances. In this work an extension of that architecture is presented for the case where millions of triplets are stored to represent the images semantic data. The idea behind this work is to speed up the search of the triples involved in pattern search. In order to fulfill this objective, several experiments are driven in Oracle 11g ORDBMS analyzing the behavior of several indices: Bitmap, B-Tree and Hash Partitioned Indexes. The conclusions obtained in this analysis are then implemented in the reference architecture.

The article is outlined as follows: in section 2 the related work is introduced, in section 3 the ORDBMS architecture is described, in section 4 the performance analysis made is presented: the indexes used, the experiments performed and the results obtained; and finally in section 5 the conclusions are included.

2 Related Work

In the last years it is possible to find in the open literature articles dealing with the integration of low-level and semantic data and also improving the efficiency recovering images by means of RDF triplets. RETIN is a search engine developed by Gony et al. [5] with the objective of diminishing the semantic gap. The approach is based on the communication with the user which is continuously asked to refine the query. The interaction with the user is composed of several binary levels used to indicate if a document belongs to a category or not.

SemRetriev by Popescu et al. [6] is a system prototype which uses an ontology in combination with CBIR (*Content Based Image Retrieval*) techniques to structure a repository of images from the Internet. Two methods are employed for recovering pictures: a) based on keywords and b) in visual similarities; in both cases the algorithm is used together with the proposed ontology.

Döller y Kosch [7] proposed and extension of an Object-Relational database to retrieve multimedia data by physical and semantic content based on the MPEG-7 standard. The main contributions of this system are: a metadata model based on MPEG-7 standard for multimedia content, a new indexation method, a query system for MPEG-7, a query optimizer and a set of libraries for internal and external applications.

The main drawbacks of the works cited before are that they are difficult to implement, they are not flexible to introduce modifications, requires certain expertise in computer graphics and the learning curve is steep.

In the work of Fletcher y Beck [8] the authors present a new indexation method to increase the joins efficiency using RDF triplets. The novelty consists on generating the index using the triple atom as an index key instead of the whole triplet. In order to access the triple they use a bucket containing pointers to them having the corresponding atom value. For example, if K is the atom value of a triple, then three buckets can be created, the first one has pointers to the triplets having the form *(K P O)*, the second with those with the form *(S K O)* y the third *(S P K)*, where *S, P* and *O* are *Subject*, *Property* and *Object* respectively. The problem with this approach is that does not take into account issues like the key or the join selectivity, which can increase the cost of recovering images in the occurrence of a high key or join selectivity.

Atre et. al. [9] introduced BitMat, which consists of a compressed bit matrix structure to store big RDF graphs. They also proposed a new method to process the joins in the query language RDF SPARQL [10]. This method employs an initial prune technique followed by a linked variable algorithm to produce the results. This allows performing bit to bit operations in queries having joins.

In the approach presented in this paper in Section 4, similar structures to the one proposed in [8] and [9] are analyzed where its implementation is performed in a simple manner using the components provided by the ORDBMS adopted.

3 Reference Architecture

The reference architecture was implemented in Oracle 11g ORDBMS, it allows the image retrieval using CBIR techniques, semantic data and the combination of both. It has a three level structure: physical (low-level), semantic (high-level) and an interface linking them.

The semantic annotations of the images are stored in triples together a reference ontology. In Fig. 1 it is shown a graph with three classes related with the property *subClassOf.* The graph and the references to the image are stored in a table. Besides, the inferred instances can be also stored as it is shown in Table 1.

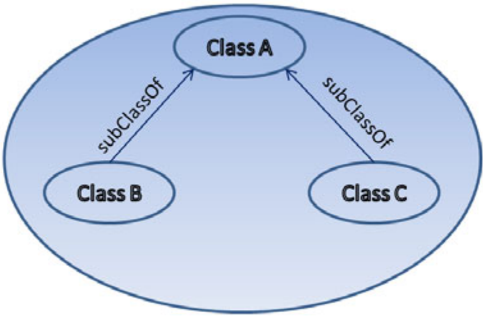


Fig. 1. RDF graph example

Table 1. RDF/RDFS triples with inferred triples (rows *i*, *k*)

row	**Subject**	**Property**	**Object**
1	Class A	Rdf:type[1]	Rdfs:Class[2]
2	Class B	Rdfs:subClassOf[3]	Class A
3	Class C	Rdfs:subClassOf	Class A
4	Image 1	Rdf:type	Class C
5	Image 2	Rdf:type	Class B
	…	…	…
i	*Image 1*	*Rdf:type*	*Class A*
k	*Image 2*	*Rdf:type*	*Class A*

The references to the images stored in the database are implemented by image OIDs (Object Identifiers), in this way Oracle assigns to each Object-Table row a unique 16 bites long OID generated by the system that permits an unambiguous object identification in distributed systems. The architecture details and its implementation can be seen in [3].

The architecture was implemented in the database by means of several UDTs (*User Defined Types*) composed of attributes and operations. These methods plays a fundamental role in recovering images, they consist of set operations allowing the combination of semantic and low-level data. The physical content and the high-level information are managed separately and then they are related using the OID obtained in the queries and the set operations: union, intersection and difference, as it is shown in Fig. 2.

In Fig. 2, *similar* is an operation defined to recovery image *OIDs* with some physical properties. Basically, the method is defined as follows: *similar(d, t)*: *SetRef*, where *d* is the physical property to employ in the search and *t* is the threshold or distance allowed respect to a reference image. This function returns the *OIDs* of the images having a lower threshold respect to the image used as a reference. The function *semResultSet(p, o): SetRef* is defined for the semantic level, where *p* is a property and *o* an object. The function returns references to images having a matching with the property an object specified. Both functions returns a set of *OIDs* (*SetRef* type) referencing images stored in an Typed-Table. Having the sets of *OIDs* is now very simple to combine and operate with them by means of the set operations:

union(SetRef, SetRef): SetRef
intersection(SetRef, SetRef): SetRef
difference(SetRef, SetRef): SetRef

Since both *similar* and *semResultSet* methods return a *SetRef* type, then any combinations of the results set is valid and can be combined in the following form:

[1] Rdf:type, is a short name of: http://www.w3.org/1999/02/22-rdf-syntax-ns#type

[2] Rdfs:Class, is a short name of: http://www.w3.org/2000/01/rdf-schema#Class

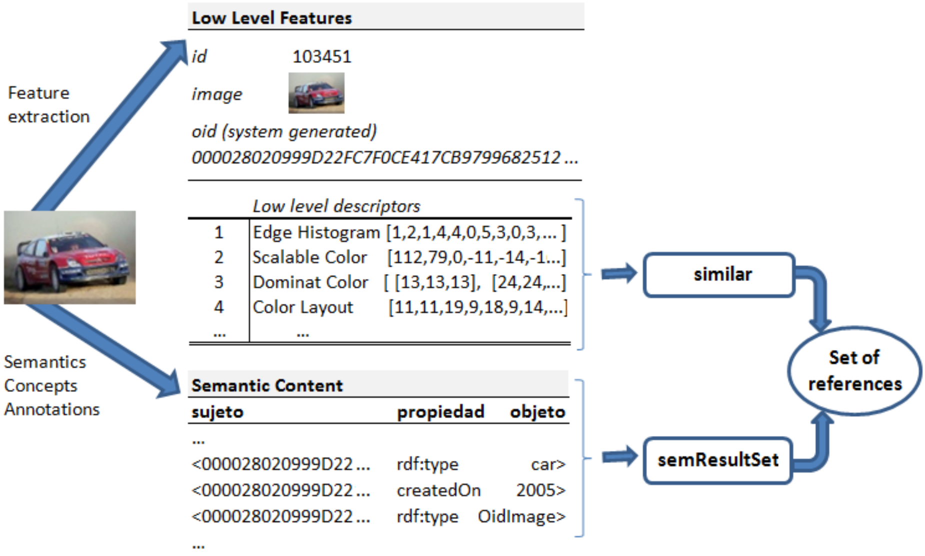
[3] Rdfs:subClassOf, is a short name of: http://www.w3.org/2000/01/rdf-schema#subClassOf

$$\boldsymbol{Op(similar\ (d_i,\ t_i),\ similar(d_j,\ t_j)):\ SetRef}$$
$$\boldsymbol{Op(semResultSet(p_n,\ o_n),\ similar(d_k,\ t_k)):\ SetRef}$$
$$\boldsymbol{Op(semResultSet(p_m,\ o_m),\ semResultSet(p_q,\ o_q)):\ SetRef}$$

where (d_i, t_i) represent descriptors and threshold respectively and (p_n, o_n) are property and object. With these operators it is possible also to pose low-level queries with different descriptors and also semantic queries having diverse patterns. Note that the functions can be used recursively and their return can be used as an input parameter to other method. In the following example, the function intersection receives as an input the results obtained in the union between *semResultSet* and *similar*, and also the result obtained in the difference of two calls to the function *semResultSet*.

$$\boldsymbol{intersection(\ union(semResultSet(p_n,\ o_n),\ \ similar(d_i,\ t_i)),}$$
$$\boldsymbol{diference(semResultSet(p_m,\ o_m),\ \ semResultSet(p_q,\ o_q)))}$$

In the next section, it is presented the study about the alternatives to improve the efficiency in the queries invoking the function *semResultSet*.



	Low level descriptors	
1	Edge Histogram	[1,2,1,4,4,0,5,3,0,3,...]
2	Scalable Color	[112,79,0,-11,-14,-1...]
3	Dominat Color	[[13,13,13], [24,24,...]
4	Color Layout	[11,11,19,9,18,9,14,...]
...	...	

Semantic Content

sujeto	propiedad	objeto
...		
<000028020999D22...	rdf:type	car>
<000028020999D22...	createdOn	2005>
<000028020999D22...	rdf:type	OidImage>
...		

Fig. 2. Physical and Semantic data representation and its relation using the OID

4 Performance Analysis Using Different Indexation Methods

4.1 Issues about Efficiency

The purpose of this work is to improve the efficiency of the reference architecture when the number of triples stored in the database is large. First it must be considered that the subject (S) is the value to find, it means that every query has the following form *(? P O)* where *P* and *O* are property and object respectively. For queries where the subject (image to recover) is the value to find are three possible search pattern options:

a. (?*s* P *?o*)
b. (*?s ?p* O)
c. (*?s* P O)

and for composed patterns the set operations are used.

The *property* attribute is employed in patterns *(?s* P *?o*) and an index is created to improve the speed of the search, for patterns (?*s* ?*p* O) the *object* attribute is employed and for (*?s* P O) the index can be generated using the attributes object and property together, or a combination of the previous individual indexes.

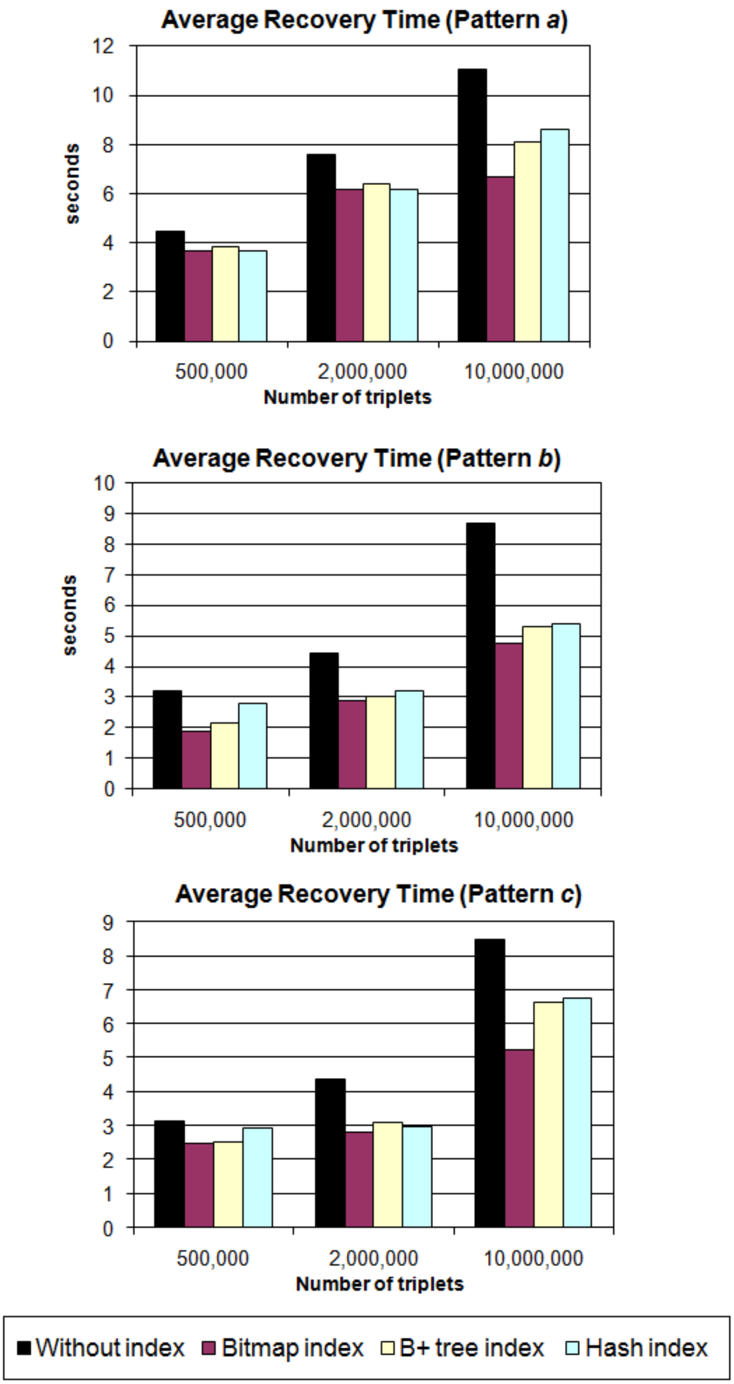
4.2 Tests Performed

For the efficiency analysis several index types are generated: Bitmap, B-Tree and Hash partitioned indexes; all of them provided in Oracle 11g DBMS. The Bitmap index was selected because it is appropriated for cases similar to the one analyzed in this article: the key has a low cardinality (high selectivity). In this structure, a bit map is constructed for each key value pointing to the block where a database register contains the data associated to the key. Other advantages of this index type are that needs lower space than traditional B-Tree indexes and some comparison operations using bits are executed faster in computer memory.

The traditional B-Tree index structure is in the opposite site of the Bitmap, so it is not appropriated for low cardinality attribute, it is used in this paper just for comparison reasons. In section 5, the results of the test show that the behavior of this structure was not so bad as was expected.

The Hash Partitioned Index is an intermediate structure where a database Table is partitioned according to an attribute selected, and a regular B-Tree index is created for each partition. The number of partition to be generated must be selected; in our case 4 partitions were created.

For the test performed, the database was loaded with different amount of triples extracted from UniProt [11]: 500,000, 2,000,000 and 10,000,000; and the average recovery time was determined using the indexes constructed. The experiments were executed on a PC CPU-INTEL CORE 2 Duo Processor 3.0 GHz, with 8 GB RAM and a 7200 rpm disk, running in Windows 2003. One hundred (100) queries were executed over the three set of triples using different selectivity values for the properties. As was explained before, selectivity counts for the number of times that the property value is repeated over the triples. The average execution times (in seconds) obtained for search pattern *a*, *b* and *c* are shown in Fig. 3.


Average Recovery Time (Pattern *a*)

Average Recovery Time (Pattern *b*)

Average Recovery Time (Pattern *c*)

Fig. 3. Average recovery time for pattern *a*, *b* and *c*

From Fig. 3 can be seen that the *bitmap index* has the better performance when the number of triplets increases. However, note that the results obtained without using an index are in the order of those employing it. In order to have an insight about this issue, a performance comparison was made between the Bitmap index and without it using triples attribute values of diverse selectivity. The results obtained can be seen in Fig. 4. From Fig. 4 it is clear that the advantage of the index diminishes when the number of triples and/or the attribute selectivity increases. This situation is very common when using a RDF graph particularly considering a property attribute. In the same direction another test was performed using *Oracle hints*, by means of this capability (hints) the query optimizer is instructed to execute the query using a specific pathway. In this case for pattern a) the average execution time was improved using the following hint:

/*+ INDEX (tripet_t ix_p) CACHE(t) */.

The first part of the hint indicates to the optimizer what index type to use and the second instructs the optimizer to place the blocks retrieved for the table at the most recently used end of the LRU (Last Recently Used) list in the buffer cache. This is particularly important for this search pattern since it is likely to make several search for the same property, for example Rdf:type. In Fig. 4 the results with the hint are shown with a green line that compared with the red one (Bitmap index without hints) can be observed the improvement in the average execution time.

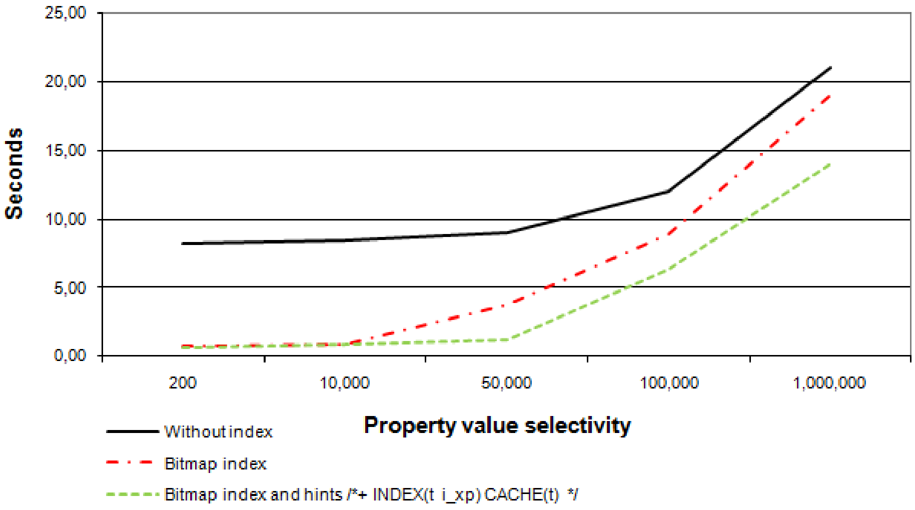


Fig. 4. Average recovery time for pattern *a*

Similar results can be obtained for pattern *b*.

In the case of pattern *c*, no improvement was obtained using the previous hints using the index generated for the composed attributes; so a test was made using the combination of the individual indexes (for property and object attributes)using the following hint:

/*+ INDEX_COMBINE(t ixp ixo) CACHE(t) */

The hints INDEX_COMBINE explicitly chooses a bitmap access path for the table. If no indexes are given as arguments for the INDEX_COMBINE hint, the optimizer uses whatever Boolean combination of bitmap indexes has the best cost estimate for the Table. If certain indexes are given as arguments, the optimizer tries to use some Boolean combination of those particular bitmap indexes by using a conjunctive (AND) bitwise operation. The results obtained are shown in Fig. 5, where again the use of the hint improves the performance.

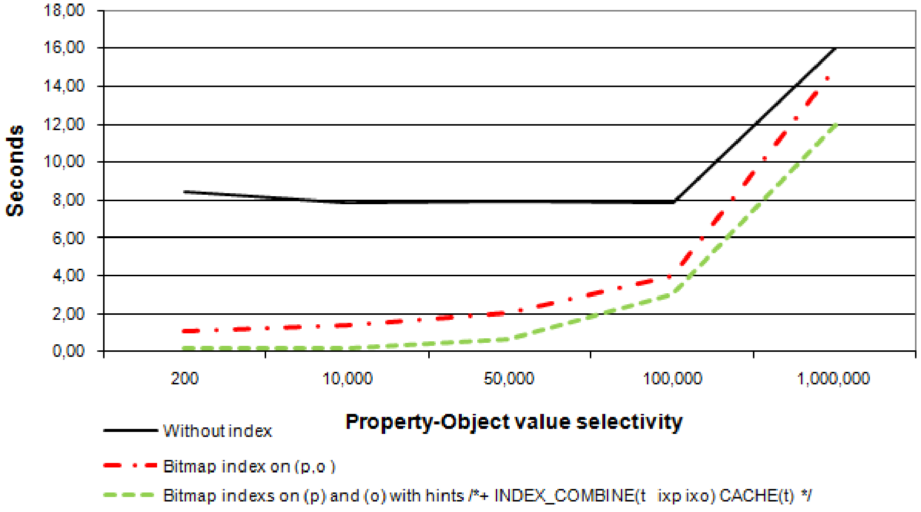


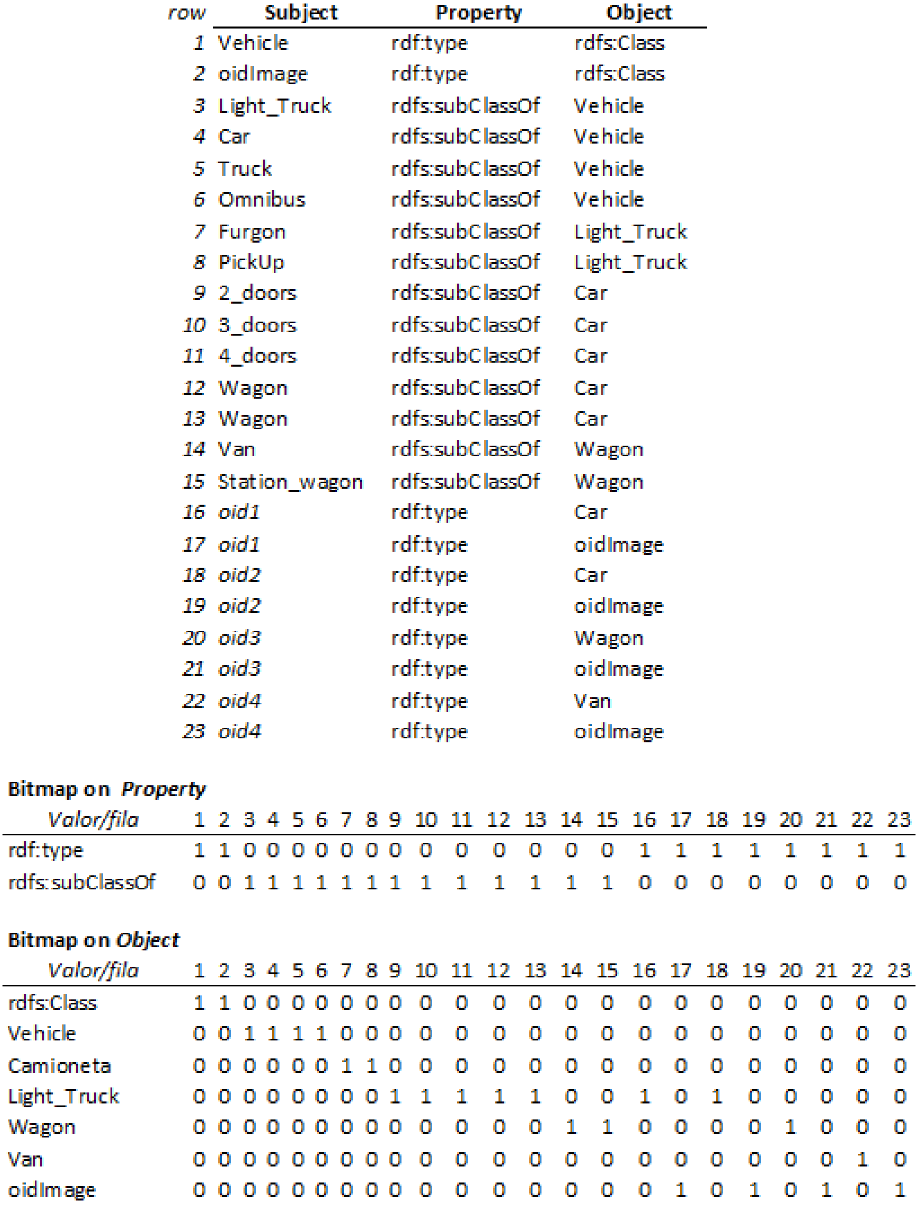
Fig. 5. Average recovery time for pattern *c*

4.3 Index Implementation in the Reference Architecture

Based on the results obtained, the implementation of a *User Defined Function* (UDF) is proposed to execute the pattern search of triples using Bitmap indexes. The function is called *search_subject*(*p*, *o*). This UDF is employed by the method *semResultSet* described in Section 3. For this purpose, an UDF similar to SEM_MATCH [12] is created but in this case this function takes the subject as a default value to search. The parameters *p* and *o* represents the property and object of the triples respectively; when the function receives a parameter with a question mark this one becomes the value to search, for example a call like *search_subject (´?p´, ´car´)* means that triples having the property ´car´ must be get.

The function parameter *p* it is just used to get the triples matching with that criteria, once having the triple, the next step is to find the subjects *(OIDs)* related to that search pattern pointing at the images stored in the database. In this sense, the pattern *(´?s´, ´rdf:type´, ´oidImage´)* must be implicitly satisfied to get the OIDs of the images.

In Fig. 6 it is shown an example about the use of the Bitmap indexes to find the triples matching with the search pattern.



row	Subject	Property	Object
1	Vehicle	rdf:type	rdfs:Class
2	oidImage	rdf:type	rdfs:Class
3	Light_Truck	rdfs:subClassOf	Vehicle
4	Car	rdfs:subClassOf	Vehicle
5	Truck	rdfs:subClassOf	Vehicle
6	Omnibus	rdfs:subClassOf	Vehicle
7	Furgon	rdfs:subClassOf	Light_Truck
8	PickUp	rdfs:subClassOf	Light_Truck
9	2_doors	rdfs:subClassOf	Car
10	3_doors	rdfs:subClassOf	Car
11	4_doors	rdfs:subClassOf	Car
12	Wagon	rdfs:subClassOf	Car
13	Wagon	rdfs:subClassOf	Car
14	Van	rdfs:subClassOf	Wagon
15	Station_wagon	rdfs:subClassOf	Wagon
16	*oid1*	rdf:type	Car
17	*oid1*	rdf:type	oidImage
18	*oid2*	rdf:type	Car
19	*oid2*	rdf:type	oidImage
20	*oid3*	rdf:type	Wagon
21	*oid3*	rdf:type	oidImage
22	*oid4*	rdf:type	Van
23	*oid4*	rdf:type	oidImage

Bitmap on *Property*

Valor/fila	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
rdf:type	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
rdfs:subClassOf	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bitmap on *Object*

Valor/fila	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
rdfs:Class	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Vehicle	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Camioneta	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Light_Truck	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0
Wagon	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0
Van	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
oidImage	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Fig. 6. The triples specification using car taxonomy and its instances are shown in the top of the figure; below, the Bitmap indexes generated with those triples

Using a query *search_subject (´?p´, O)* the Bitmap index created for *object* is employed. For example the query *search_subject (´?p´, 'Car')* retrieves the rows 10-13, 16 and 18, then only the subjects of those rows must be taken into account; but not all of them are included in the final results only those satisfying the pattern *(´?s´, ´rdf:type´, ´oidImage´)* because they have OIDs values referencing images in the database.

For a query of type *search_subject(P, ´?o´)* it is used the index created for the *property* column. For example the query *search_subject(´Rdf:type´, ´?o´)* get the rows 1, 2, 16-23, then these rows must be intersected with subjects having the pattern *(´?s´, ´rdf:type´, ´oidImage´).*

Finally, when the query has the form *search_subject (P, O)* the intersection of the Bitmap indexes over the columns *property* and *object* must be used. For example, the query *search_subject (´Rdf:type´, 'Car')*having the map property recovers the rows 1, 2, 16-23 and with the bitmap index *object* rows 10-13, 16,18 are obtained. The intersection are the rows 16 and 18; the subject of those rows are intersected with the subject of the pattern *(´?s´, ´rdf:type´, ´oidImage´)* to get the images in the final result.

5 Conclusions

In this work it is presented a performance analysis for recovering semantic data stored in an Object-Relational database in the form of RDF triples. Different indexation methods are selected to perform the analysis. The triples are used to relate images with its semantic information via the OIDs created by the ORDBMS when the image is stored in a Typed-Table. The goal pursued with the use different indexation methods is to improve the efficiency in recovering the image via a faster retrieve of the OIDs. A Reference Architecture was employed to drive the test and also to implement the results obtained.

One conclusion arrived in this work indicates that the Bitmap index has a better performance compared to the B-tree and Hash Partitioned indexes when the RDF graph is composed of thousands and millions of triples. All the experiments were executed using Oracle 11g ORDBMS. Another conclusion verified was that the combination of two individual Bitmap indexes has a better performance than the composed one over property and object columns. The use of hints may improve the efficiency when used appropriately.

Based on the previous conclusions, the Bitmap index together with the *search_subject* UDF function were implemented to speed up the RDF triples search and as a consequence the image recovery. It is important to note that the architecture, the index and functions used, are all implemented using tools that are provided by most of the nowadays commercial ORDBMS, which facilitates its realization.

References

1. Neumamm, D., Gegenfurtner, K.: Image Retrieval and Perceptual Similarity. ACM Transactions on Applied Perception 3(1), 31–47 (2006)
2. Alvez, C., Vecchietti, A.: A model for similarity image search based on object-relational database. IV Congresso da Academia Trinacional de Ciências, 7 a 9 de Outubro de 2009 - Foz do Iguaçu - Paraná / Brasil (2009)

3. Alvez, C.E., Vecchietti, A.R.: Combining Semantic and Content Based Image Retrieval in ORDBMS. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6277, pp. 44–53. Springer, Heidelberg (2010)
4. Jim, M.: (ISO-ANSI Working Draft) Foundation (SQL/Foundation). ISO/IEC 9075-2:2003 (E), United States of America, ANSI (2003)
5. Gony, J., Cord, M., Philipp-Foliguet, S., Philippe, H.: RETIN: a Smart Interactive Digital Media Retrieval System. In: ACM Sixth International Conference on Image and Video Retrieval CIVR 2007, Amsterdam, The Netherlands, July 9-11, pp. 93–96 (2007)
6. Popescu, A., Moellic, P.A., Millet, C.: SemRetriev – an Ontology Driven Image Retrieval System. In: ACM Sixth International Conference on Image and Video Retrieval CIVR 2007, Amsterdam, The Netherlands, July 9-11, pp. 113–116 (2007)
7. Döller, M., Kosch, H.: The MPEG-7 Multimedia Database System (MPEG-7 MMDB). The Journal of Systems and Software 81, 1559–1580 (2008)
8. George, H.L., Fletcher, P.W.: Beck: Scalable indexing of RDF graphs for efficient join processing. In: ACM Conference on Information and Knowledge Management CIKM 2009, pp. 1513–1516 (2009)
9. Atre, M., Chaoji, V., Zaki, M.J., Hendler, J.A.: Matrix "Bit"loaded: A Scalable Lightweight Join Query Processor for RDF Data International World Wide Web Conference Committee (IW3C2), April 26-30. ACM, Raleigh (2010)
10. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation (January 15, 2008)
11. UniProt RDF, `http://dev.isb-sib.ch/projects/uniprot-rdf/`
12. Chong, E.I., Das, S., Eadon, G., Srinivasan, J.: An efficient SQL-based RDF querying scheme. In: Proceedings of the 31st international conference on Very large data bases, VLDB 2005, Trondheim, Norway, pp. 1216–1227 (2005)

Automatic Identification of Web Query Interfaces

Heidy M. Marin-Castro, Victor J. Sosa-Sosa, and Ivan Lopez-Arevalo

Center of Research and Advanced Studies of the National Polytechnic Institute
Information Technology Laboratory
Scientific and Technological Park of Tamaulipas TECNOTAM
{hmarin,vjsosa,ilopez}@tamps.cinvestav.mx

Abstract. The amount of information contained in databases in the Web has grown explosively in the last years. This information, known as the Deep Web, is dynamically obtained from specific queries to these databases through Web Query Interfaces (WQIs). The problem of finding and accessing databases in the Web is a great challenge due to the Web sites are very dynamic and the information existing is heterogeneous. Therefore, it is necessary to create efficient mechanisms to access, extract and integrate information contained in databases in the Web. Since WQIs are the only means to access databases in the Web, the automatic identification of WQIs plays an important role facilitating traditional search engines to increase the coverage and access interesting information not available on the indexable Web. In this paper we present a strategy for automatic identification of WQIs using supervised learning and making an adequate selection and extraction of HTML elements in the WQIs to form the training set. We present two experimental tests over a corpora of HTML forms considering positive and negative examples. Our proposed strategy achieves better accuracy than previous works reported in the literature.

Keywords: Deep Web, Databases, Web query interfaces, classification, information extraction.

1 Introduction

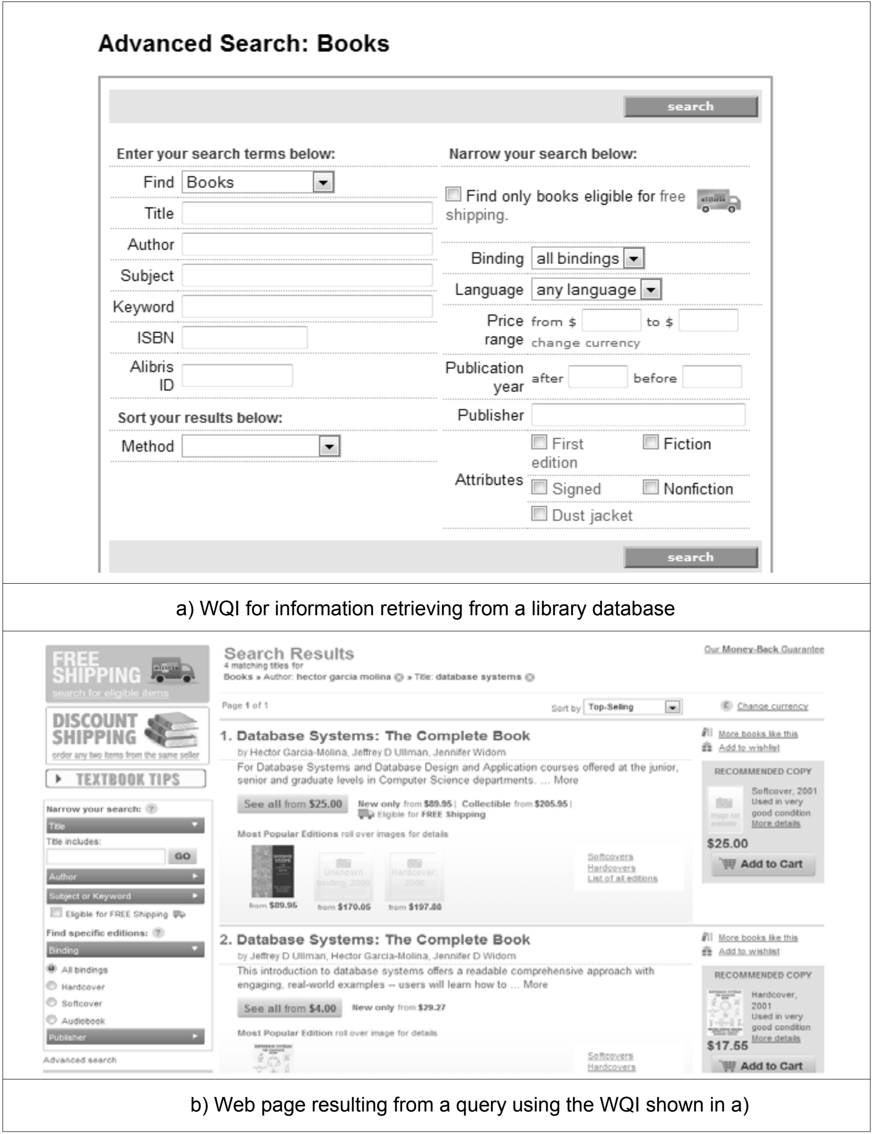
In recent years, the explosive growth of the Internet has made the Web to become one of the most important sources of information and currently a large number of databases are available through the Web. As a consequence, the Web has become dependent of the vast amount of information stored in databases on the Web. Unlike the information contained in the Indexable Web [4] that can be easily accessed through an analysis of hyperlinks, matches with keywords or other mechanisms implemented by some search engine, the information contained in the databases on the Web can only be accessed via Web Query Interfaces (WQIs) [4]. We define WQI as an HTML form that is intended for users that want to query a database on the Web.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 297–306, 2011.

Given the dynamic nature of the Web, new Web pages are aggregated constantly and some others are removed or modified. This makes that the automatic discovery of WQIs that serve as entry points to the databases on the Web be a great challenge. Moreover, most of the HTML forms contained in Web pages are not used for querying databases in the Web, such as HTML forms for discussion groups, logging, mailing list subscriptions, online shopping, among others.

The design of WQIs is heterogeneous in its content, presentation style and query capabilities, which makes more complex the automatic identification of information contained in these interfaces. The WQIs are formed by HTML elements *(selection list, text input box, radio button and checkbox, etc.)* and fields for these elements. A field has three basic properties: *name*, *label* and *domain*. The property *name* corresponds to the name of the field, *label* is the string associated with the field in the WQI or the empty string in case the *label* is not associated with the field, the *domain* is the set of valid values that the field can take [13]. The fields are associated to the HTML elements and these are related to form a group. Various groups form a super-group producing as a result a hierarchical structure of the WQI. A property that characterizes the WQIs is their semi-structured content. This makes the WQIs different to Web pages that reside in the Indexable Web which content is not structured information [13]. An example of a WQI to search books is shown in figure 1. This WQI is used to generate dynamically Web pages, as the one shown in figure 1 b).

In this work we present a strategy for automatic identification of WQIs using supervised learning. The key part in this strategy is to make an adequate selection of HTML elements that allow to determine if a Web page contains or not a WQI. Several works reported in the literature for identification of WQIs [5], [13], [14] have not provided a detailed study of design, internal structural, number and the type of HTML elements of WQIs that can be taken as reference for identification of WQIs. In this work we use features contained in HTML forms, like HTML elements, and corresponding fields, to form *characteristic vectors* used in the classification task. These features are extracted without considering a specific domain of the application. The extraction process of features is challenging because the WQIs lack a formal specification and they are developed independently. Moreover, the majority of WQIs are designed with the markup language HTML, which does not express data structures and semantics. There exists some works that have dealt with the automatic identification of WQIs, for example [5], [3], [13], [14] among others. However, it is not validated if the WQIs identified for these works really allow to get information from databases in the Web. In [3], the authors consider some features similar to the ones we use in this work. However, they do not use “select” and “combo-boxes” HTML elements, which contribute with more information for the identification of WQIs. In addition, the majority these related works try to identify WQIs for specific domains, which limits the application of those strategies to different contexts.



a) WQI for information retrieving from a library database

Search Results
4 matching titles for
Books » Author: hector garcia molina » Title: database systems

Page 1 of 1

1. Database Systems: The Complete Book
by Hector Garcia-Molina, Jeffrey D Ullman, Jennifer Widom
For Database Systems and Database Design and Application courses offered at the junior, senior and graduate levels in Computer Science departments. ... More
See all from $25.00 New only from $89.95 | Collectible from $205.95 |
Eligible for FREE Shipping
Most Popular Editions roll over images for details
from $89.95 from $170.05 from $197.88
RECOMMENDED COPY
Softcover, 2001
Used in very good condition
More details
$25.00

2. Database Systems: The Complete Book
by Jeffrey D Ullman, Hector Garcia-Molina, Jennifer D Widom
This introduction to database systems offers a readable comprehensive approach with engaging, real-world examples -- users will learn how to ... More
See all from $4.00 New only from $29.27
Most Popular Edition roll over image for details
RECOMMENDED COPY
Hardcover, 2001
Used in very good condition
More details
$17.55

b) Web page resulting from a query using the WQI shown in a)

Fig. 1. An example of a WQI

The rest of the paper is organized as follows. Section 2 briefly describes some of the works related to the identification of WQIs. Section 3 introduces our strategy for automatic identification of WQIs. Section 4 describes the experimental results performed. Finally, section 5 present a summary of this work.

2 Related Work

The first challenge for modeling and integrate databases in the Web is to extract and understand the content of the WQIs and their capabilities for querying that they support.

In [2], the authors propose a new strategy called Hierarchical Form Identification (HIFI). This strategy is based on the decomposition of the space of the HTML Form features and uses learning classifiers, which are the best suited for this kind of application. That work uses a focused crawler that uses the characteristics of the Web pages that it identifies as WQIs to focus the searching on a specific topic. The crawler uses two classifiers to guide its search: a generic classifier and a specialized classifier. The generic classifier allows to eliminate HTML forms that not generate any query into databases in the Web. The specific classifier identifies the domain of the HTML forms selected by the generic classifier. The decomposition of the characteristic space uses a hierarchy of form types through the selected HTML forms followed by a analysis of WQIs related to a specific domain. The authors used structural patterns to determine whether a Web page is or not a WQI. They observed empirically that the structural characteristics of an HTML form can determine whether the form is or not a WQI. In addition, their specialized classifier uses the textual content of an HTML form to determine its domain. For this task, they use the C4.5 and Support Vector Machine (SVM) classification algorithms [9].

In [14], Zhang, et. al. hypothesize the existence of a hidden syntax that guides the creation of query interfaces from different sources. Such hypothesis allows to transform query interfaces into a visual language. In that work authors stated that the automatic extraction task is essential to understand the content of a WQI. This task is rather "heuristic" in nature so it is difficult to group pairs of the closest elements by spatial proximity or semantic labeling in HTML forms. One proposed solution for this problem is the creation of a 2P grammar. The 2P grammar allows to identify not only patterns in the WQIs but also in their precedence. This grammar is over sized with more that 80 productions than were manually derived from a corpora with 150 interfaces.

Others works to represent the content of the WQIs are based in the use of a hierarchical schema trying to capture the semantic part of the fields and HTML elements in an interface as much as possible [7], [5] and [13]. However, these works can not identify completely if a Web page is or not a WQI. Therefore the identification, characterization and classification of WQIs continue been a challenging research topic.

Table 1 shows the level of accuracy of representative works for automatic identification of WQIs. These works uses visual analysis of characteristics of the Web pages tested and heuristic techniques based on textual properties such as the number of words, similarity between words, etc., as well as schema properties such as position of a component, distance among components, etc. However, the most of these works present the following disadvantages:

- The human intervention is constantly required to perform the identification of WQIs
- Their approach is to determine the domain of the WQIs without performing the automatic identification of WQIs
- Lack of a clear, precise and defined scheme for automatic identification of WQIs

Table 1. Reported works in the literature for identification and characterization of Web query interfaces

Ref.	Technique	Accuracy
[2]	Hierarchical decomposition of characteristic space (HIFI)	90%
[5]	Automatic generation of features based on a limited set of HTML tags	85%
[13]	Bridging Effect	88%
[14]	Grammar 2P and tree parse	85%

In the next section we describe with detail our proposed strategy for the identification of WQIs.

3 Proposed Strategy

The proposed strategy for the identification of WQIs is composed of three phases: a) searching of HTML Forms in Web pages, b) automatic extraction of HTML elements from HTML forms and c) automatic classification of HTML forms. In the first phase we collected automatically a set of Web pages by using a Web crawler rejecting other type of documents (pdf, word, pps, etc). Then, we searched into the internal structure of the Web pages for the presence of forms to delimitate the search space. In the second phase we built a extractor program that obtains the number of occurrences of HTML elements in the forms and the existence of strings or keywords *(search, post or get)* independently of the domain. Finally in the third phase we built a training set to classify HTML forms in WQIs.

The implementation of our proposed strategy is described in algorithm 1. We begin with a set W of Web pages (containing WQIs) from the UIUC repository [1] and a set N of Web pages (without WQIs) that were manually obtained. We also count the number of occurrences of each HTML element with the aim of forming a characteristic vector that allows to classify Web pages to determine if they contain or not WQIs. The output of the implementation is a text file containing the number of occurrences of each HTML element as well as the values of true or false in relation to the existence of the strings *get*, *post* y *search* in each set. This file serves as input to the classifiers (Naive Bayes, J48 or SVM), which determine the class of each URL, in this case if it is a WQI or not.

The automatic extraction of HTML elements is based on the use of HTML Parser Jericho [6], and the classification of HTML forms uses structural features to eliminate HTML forms that do not represent a WQI.

Algorithm 1. Automatic Identification of WQI

```
Require: $W$: Web pages (WQIs), $N$: Web pages (Not WQIs), $Extractor$: Jericho,
    $classifier$: Naive Bayes, J48 and SVM,
Ensure: Output: Instances classified as WQIs or Not WQIs
Search keywords: $< form > < /form >$ in $W$ and $N$
if EXIST(keywords) then
    Table = $label < String, Integer >$
    Definition of HTML labels "$select$", "$button$", "$text$",
    "$checkbox$", "$hidden$", "$radio$", "$file$",
    "$image$", "$submit$", "$password$", "$reset$"
    "$search$", "$post$", "$get$"
    for $HTMLsegment$ from $< form >$ to $< /form >$ do
        Call to labels = Extractor($HTML\ segment$)
        if ($label = definite\ label$) then
            $Table = Table(label, counter + 1)$
        end if
    end for
    file=$< Table(label, counter), tag >$
end if
Classify($classifier$,$file$)
```

4 Experimental Results

This section describes the effectiveness of the proposed strategy using HTML positive forms (WQIs) and HTML negative forms (HTML forms that do not generate queries to a database on the Web: logging forms, discussion interfaces groups, HTML subscription list of mail, shopping forms in markets online, etc.). In order to show the effectiveness of the strategy in the identification of WQIs, the precision rate was calculated using the Naive Bayes, J48 and SVM algorithms (using the Sequential Minimal Optimization (SMO) algorithm at various degrees of complexity) to classify HTML forms in positive or negative.

To carry out the test, two corpora of HTML forms were built with positive and negative examples. In the first corpus, 223 WQIs from the database TEL-8 Query Interfaces [8] from the UIUC repository [1] were used as positive examples and 133 negative examples that were manually obtained. The next 14 features were extracted: number of images , number of buttons, number of input files, number of *select* labels, number of *submit* labels, number of *textboxes*, number of *hidden* labels, number of *resets* labels, number of *radio* labels, number of *checkboxes*, number of *password* and the presence of the strings *get*, *post* y *search*. In the second corpus, 22 WQIs of the database ICQ Query [11] Interfaces from the

UIUC repository [1] were used as positive examples and 57 negative examples that were gathered manually.

During the learning task, the predictive model was evaluated for the two corpora using the *10-fold cross validation* technique [12], which divides randomly the original sample of data in 10 sub-sets of (approximately) the same size. From the 10 sub-sets a single subset is kept as the validation data for testing the model and the $K - 1$ sub-sets remaining are used as training data. The *cross-validation* process is repeated 10 times (the folds), with each of the 10 sub-sets used exactly once as validation data. The average of results in the 10 folds is obtained to produce a single estimate. The advantage of cross validation is that all the observations are used for training and validation.

We used three algorithms for classification of positive and negative HTML forms using $Weka$ [10]: Naive Bayes, J48 and SMO of SVM. These algorithms are considerate as three of the best algorithms for classification with high accuracy [10]. The classification results for both corpora of HTML forms are displayed in table 2.

Table 2. Accuracy of the classification algorithms with the two corpora built

Classifier	**Corpus**	**Accuracy**
Naive Bayes	1 (356 instances)	98.59%
J48	1 (356 instances)	98.87%
SMO (complexity parameter=1)	1 (356 instances)	91.57%
SMO (complexity parameter=2)	1 (356 instances)	94.38%
SMO (complexity parameter=3)	1 (356 instances)	94.34%
NaiveBayes	2 (79 instances)	93.67%
J48	2 (79 instances)	97.46%
SMO (complexity parameter=1)	2 (79 instances)	88.60%
SMO (complexity parameter=2)	2 (79 instances)	88.60%
SMO (complexity parameter=3)	2 (79 instances)	89.87%

As it can be seen in table 2, the classification algorithm J48 obtains the best accuracy compared with the other two algorithms because J48 works fine in presence of not relevant attributes [10].

In a related work [2], the authors use the same repository TEL-8 Query Interfaces [8] than the presented in this paper. Also, they use the classification algorithms Naive Bayes, J48, Multilayer perceptron and SVM to classify WQIs. However, the level of accuracy that they obtained for the identification of WQIs is lower compared to the results that we achieved.

This is because their strategy for selection and extraction of HTML elements is not complete. The strategy proposed in [2] does not present a detail analysis of the HTML elements that contribute with more information about the identification of WQIs. For example, consider the WQIs shown in figure 2 related to airline flights, the HTML elements predominant in these WQIs are *selection lists*, *checkboxes* and the existence of one single button to perform the searching

Table 3. Comparison of accuracy obtained by the Naive Bayes, J48 and SVM classifiers

Works	No. examples	Naive Bayes	J48	SVM (degree=1)	SVM (degree=2)	SVM (degree=3)
Barbosa and Freire [2]	216 WQIs 259 not WQIs	76%	90.95%	85.1%	83.8%	85.1 %
Test 1	223 WQIs 133 not WQIs	98.59%	98.87%	91.57%	94.38%	93.34 %
Test 2	22 WQIs 57 not WQIs	93.67%	97.46%	88.60%	88.60%	89.87 %

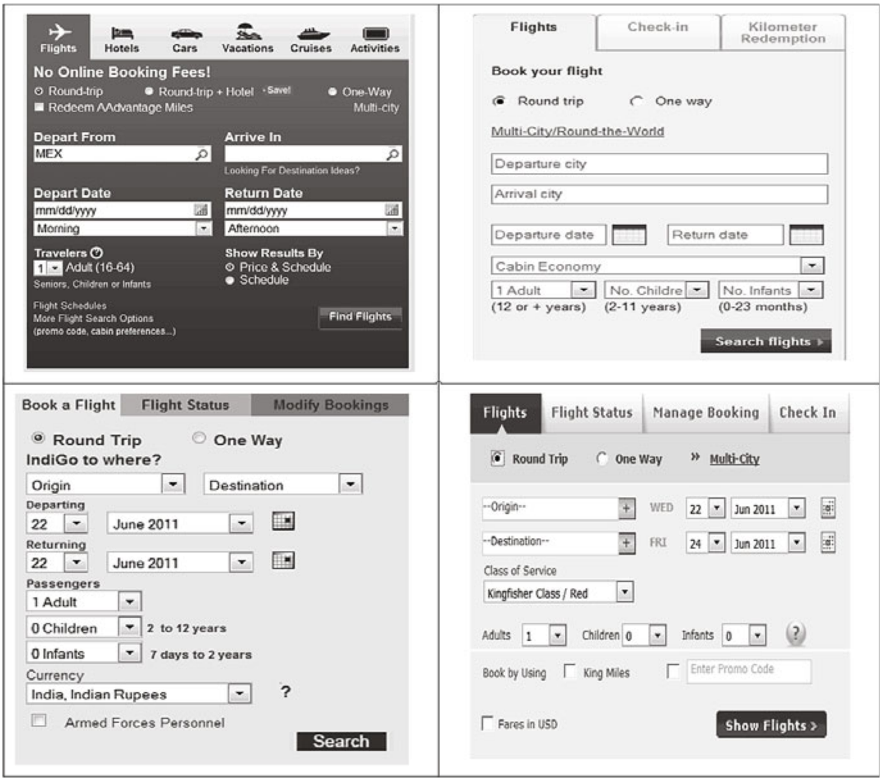


Fig. 2. Example of WQIs of airlines-flights

of flights. This button can contains a string related to the string "search" such as *Find Flights*, *Search Flights*, *Show Flights*, *Search*, *Go*, among others. Therefore, in our proposed strategy we considerate HTML elements that contribute more with the classify Web pages in WQIs and not WQIs. Table 3 shows the percent of accuracy obtained by the proposed strategy carrying out two tests, using the corpora mentioned above. The results are compared with those presented in [2] for WQIs automatic identification.

5 Conclusions

Web Query Interfaces (WQIs) allow to access databases in the Web and retrieve information that is not reachable by traditional search engines. The automatic identification of WQIs is a complex task given the heterogeneity in their design, style, semantic content among others. This paper presented a new strategy for automatic identification and classification of WQIs, its performance was evaluated using two corpora of HTML forms. The proposed strategy uses supervised classification to determine if an HTML form is or not a WQI. The supervised classifier uses characteristic vectors that contain information about HTML elements of forms contained in Web pages as the number of *textboxes*, command *buttons*, labels, the presence of keywords as “submit”, “search”, “get”, or “post”, etc. We obtained better level of accuracy compared with others reported works, the improvement obtained with the proposed strategy is mainly given by the use of HTML elements that are not directly associated with a specific domain of interest.

References

1. The UIUC web integration repository. Computer Science Department, University of Illinois at Urbana-Champaign (2003), http://metaquerier.cs.uiuc.edu/repository
2. Barbosa, L., Freire, J.: Searching for hidden-web databases. In: Proceedings of the 8th ACM SIGMOD International Workshop on Web and Databases, Baltimore, Maryland, USA, pp. 1–6 (2005)
3. Barbosa, L., Freire, J.: Combining classifiers to identify online databases. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, pp. 431–440. ACM, New York (2007), http://doi.acm.org/10.1145/1242572.1242631
4. Bergman, M.K.: The deep web: Surfacing hidden value (white paper). Journal of Electronic Publishing 7(1) (2001)
5. Cope, J., Craswell, N., Hawking, D.: Automated discovery of search interfaces on the web. In: Proceedings of the 14th Australasian Database Conference ADC 2003, vol. 17, pp. 181–189. Australian Computer Society, Inc., Darlinghurst (2003), http://portal.acm.org/citation.cfm?id=820085.820120
6. Jericho HTML Parser: A Java Library for parsing HTML documents. Sourceforge Project (2010), http://jericho.htmlparser.net/docs/index.html (last accessed December 2010)
7. Kabisch, T., Dragut, E.C., Yu, C.T., Leser, U.: A hierarchical approach to model web query interfaces for web source integration. PVLDB 2(1), 325–336 (2009)
8. Chang, K.C.-C., He, B., Li, C., Zhang, Z.: TEL-8 Query Interfaces. UIUC Web Integration Repository (2003), http://metaquerier.cs.uiuc.edu/repository/datasets/tel-8/index.html (last accessed June 2011)

9. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
10. Weka Machine Learning Project: Weka, http://www.cs.waikato.ac.nz/~ml/weka
11. Wu, W., Yu, C., Doan, A., Meng, W.: ICQ Query Interfaces. UIUC Web Integration Repository (2003), http://metaquerier.cs.uiuc.edu/repository/datasets/icq/index.html (last accessed June 2011)
12. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Academic Press, USA (2000)
13. Wu, W., Yu, C., Doan, A., Meng, W.: An interactive clustering-based approach to integrating source query interfaces on the deep Web. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD 2004, pp. 95–106. ACM, New York (2004), http://doi.acm.org/10.1145/1007568.1007582
14. Zhang, Z., He, B., Chang, K.C.C.: Understanding Web query interfaces: best-effort parsing with hidden syntax. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD 2004, pp. 107–118. ACM, New York (2004), http://doi.acm.org/10.1145/1007568.1007583

A GRASP with Strategic Oscillation for a Commercial Territory Design Problem with a Routing Budget Constraint

Roger Z. Ríos-Mercado[1] and Juan C. Salazar-Acosta[2]

[1] Universidad Autónoma de Nuevo León, Graduate Program in Systems Engineering,
AP 111–F, Cd. Universitaria, San Nicolás de los Garza, NL 66450, Mexico
roger.rios@uanl.edu.mx

[2] Aleph5, Monterrey NL, Mexico
juan.salazar@aleph5.com

Abstract. This paper addresses a commercial districting problem arising in the bottled beverage distribution industry. The problem consists of grouping a set of city blocks into territories so as to maximize territory compactness. As planning requirements, the grouping seeks to balance both number of customers and product demand across territories, maintain connectivity of territories, and limit the total cost of routing. A combinatorial optimization model for this problem is introduced. Work on commercial territory design has particularly focused on design decisions. This work is, to the best of our knowledge, the first to address both design and routing decisions simultaneously by considering a budget constraint on the total routing cost in commercial territory design. A greedy randomized adaptive search procedure that incorporates advanced features such as adaptive memory and strategic oscillation is developed. Empirical evidence over a wide set of randomly generated instances based on real-world data show a very positive impact of these advanced components. Solution quality is significantly improved as well.

Keywords: Combinatorial optimization, Territory design, GRASP, Strategic oscillation, Adaptive memory.

1 Introduction

Territory design consists of grouping small geographical units into larger clusters called territories for the purpose of making more manageable the entire set. Current applications are vast: political districting (Bozkaya, Erkut, and Laporte [2]), school districting (Caro et al. [3]), and recollection of waste electric and electronic equipment (Fernández et al. [6]), to name a few. A more comprehensive survey on territory design can be found in the work of Kalcsics, Nickel, and M. Schröder [10], and Duque, Ramos, and Suriñach [4].

This paper deals with a commercial territory design problem (CTDP) motivated by a real-world application in a beverage distribution firm. The firm seeks

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 307–318, 2011.

to form territories that are as compact as possible subject to planning criteria such as territory balance with respect to two different activities (number of customers and product demand), territory connectivity, and unique assignment.

The CTDP problem was introduced by Ríos-Mercado and Fernández [11]. Other variations of this problem have been studied recently [12,13]. In each of these works, a dispersion measure for assessing territory compactness based on Euclidean distances is considered. In this regard, a good or desirable districting plan is one with low values of this dispersion measure. This is similar to certain location problems such as the p-Median Problem and the p-Center Problem where a dispersion based measure is minimized. It is clear that in real world applications, network-based distances are more representative of distances between basic units. In this regard, in many cases one can use models based on Euclidean distances without loss of generality because one can always replace Euclidean distances by their corresponding shortest path distances. This is also true in TDP applications where connectivity constraints are not considered. When connectivity constraints are taken into account, the fact that the territory dispersion measure is limited to those paths entirely contained in the territory leads to very intractable models as the distances between basic units end up being solution-dependent. Therefore, in previous work on TDPs with connectivity constraints authors have addressed the problems under the assumption of the Euclidean-based distances to make the problems more tractable. It is easy to find examples where a shortest path between two nodes in a given territory falls outside the territory. In addition, a single budget constraint on the total routing cost is introduced in our problem. These two aspects (network-based distances and routing cost) have been often neglected in previous work on commercial territory design and it represents a more challenging problem. Among the few works addressing the design and routing simultaneously is the one of Haugland, Ho, and Laporte [9] but in a different application. To the best of our knowledge, our work is the first to address these issues in this class of TDPs. Our work can be seen as extension of the work by Ríos-Mercado and Fernández [11], where a routing cost constraint is introduced, and network-based distances are explicitly considered.

To address this problem, a Greedy Randomized Adaptive Search Procedure (GRASP) with adaptive memory and strategic oscillation is proposed and evaluated over a range of randomly generated instances based on real-world data. GRASP [5] is a metaheuristic that has been widely used to solve a large number of combinatorial optimization problems. Recent successfull applications of GRASP can be found on recycling and recollection [6], including commercial territory design [11,13]. The proposed iterative procedure is formed by three steps (two in a construction phase and one in an improvement phase). First a partial territory design, where not all units have been assigned to territories, is built by taking into account the dispersion-based objective function. In the second step, the remaining unassigned units are assigned to territories by using a merit function that weights both the objective function and the violation of the balancing constraints. This merit function includes a frequency-based term as an adaptive memory mechanism. In this construction phase the routing constraint is relaxed. After a solution is built, routing costs are evaluated by solving a small-scale Traveling Salesman Problem with Multiple Visits (TSPM) [8] within each territory including the distribution center. In the post-processing phase, the procedure attempts to improve the objective function and to satisfy the violated balancing and routing constraints. To this end, strategic oscillation is performed. Strategic oscillation is a very powerful technique [7] that consists in relaxing some constraints and add them to the objective function with a penalty parameter. This penalty parameter is dynamically updated throughout the execution of the algorithm. This dynamic update allows the search trajectory to oscillate between the feasible and the infeasible space (those solutions not satisfying these relaxed constraints). The motivation for this comes from the fact that, by allowing the problem to become temporarily unfeasible, it is possible to visit solution regions that otherwise would be impossible to explore. The results are very encouraging. Empirical evidence indicates the great impact of the proposed advanced components, particularly the use of strategic oscillation within the local search.

The rest of the paper is organized as follows. In Section 2, the problem description and its corresponding combinatorial model are presented. Section 3 describes the proposed approach. The empirical work is discussed in Section 4. Final remarks are drawn in Section 5.

2 Problem Description

Each node or basic unit (BU) represents a city block which might contain one or more customers. Every edge represents adjacency between two BUs. Two BUs are said to be adjacent if their associated blocks share a positive portion of a street. Let $G = (V, E)$ be an undirected and planar graph that represents the city map, where V is the set of all BUs (nodes) and E is the set of edges, representing the adjacency between BUs. Let $G = (V, E)$ be an undirected and planar graph that represents the distribution network, where V is the set of all BUs (nodes) and E is the set of edges, representing the connectivity between BUs. Let d_{ij} be the Euclidean distance between BUs i and j, $(i, j) \in E$. The subset $X_k \subseteq V$ represents a territory. Every node i has two different activities associated to it: number of customers (measured by parameter w_i^1) and total demand (measured by parameter w_i^2). The size of territory X_k with respect to activity a is given as follows: $w^a(X_k) = \sum_{i \in X_k} w_i^a$, $a \in A = \{1, 2\}$. The perfect measure of a territory with respect to activity a is defined by $\mu^a = w^a(V)/p$, where p is the known number of territories. Because of the discrete nature of the problem, it is almost impossible to obtain this perfect measure for each activity. To cope with this, the company allows a tolerance deviation (represented by τ^a) from the average value μ^a. Let T_{ij}^S be the shortest path from node i to node j in $G^S = (S, E(S))$, the subgraph of G induced by set S, $S \subseteq V$, $i, j \in V$ with corresponding length t_{ij}^S. To model the dispersion of a given design $X = (X_1, \dots, X_p)$ we use the diameter-based measure given by: $f(X_1, X_2, \dots, X_p) = \max_{k=1,\dots,p} \left\{ \max_{i,j \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\}$.

It is required that each territory be connected, i.e., one should be able to traverse a single territory without having to pass through other territories.

Let Π be the collection of all possible p-partitions of V, and let $X = (X_1, \ldots, X_p)$ be an arbitrary partition $X \in \Pi$. The combinatorial optimization TDP model is shown below:

$$\min_{X \in \Pi} f(X) = \max_{k=1:p} \left\{ \max_{ij \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\} \tag{1}$$

subject to

$$w^a(X_k) \in \left[(1-\tau^a)\mu^a, (1+\tau^a)\mu^a\right], \; a = 1,2 \tag{2}$$

$$G^{X_k} = (X_k, E(X_k)) \text{ must be connected, } k = 1, 2, \ldots, p \tag{3}$$

$$\sum_{k=1,\ldots,p} R(X_k) \leq C \tag{4}$$

In the objective we use a diameter-based function (1) to measure territory dispersion. Constraints (2) assure that each territory is appropriately balanced. The requirement of territory connectivity is represented by (3), which means that, for every territory X_k, there exists a path from every node in the territory to every other node in the territory. Constraint (4) imposes a limit (denoted by C) on the total traveling cost, where $R(X_k)$ represents the routing cost of territory X_k. Assuming the distribution center is represented by node 0, $R(X_k)$ can be seen as the routing cost incurred when traversing the nodes in X_k from the distribution center and returing to it. The problem of finding a closed walk of minimum length where each node is visited at least once is known as the Traveling Salesman Problem with Multiple Visits (TSPM) [8]. This problem can be transformed into a TSP by replacing the edge cost with the shortest path distances in G. In the absence of negative cycles, such as our problem, shortest path distances between all pairs of nodes of a graph can be computed using efficient algorithms. Thus, $R(X_k) = \text{TSPM } (X_k \cup \{0\}) = \text{TSP}(X_k \cup \{0\})$, where the TSP operates in a graph that uses shortest path distances $t_{ij}^{X_k}$ instead of d_{ij}. Given that in our particular problem, each individual territory has approximately 25-40 BUs in the worst case, computing $R(X_k)$ is reduced to solving a TSP with 30-40 cities. This can be computed exactly very efficiently by state-of-the-art branch-and-cut methods. In our case, we use CONCORDE [1].

3 Proposed Algorithm

GRASP [5] is a metaheuristic that has been widely used to solve a large number of combinatorial optimization problems. Recent successfull applications of GRASP can be found on recycling and recollection [6], including commercial territory design [11,13]. In each GRASP iteration there are two phases: construction and post-processing. Construction phase aims at building a feasible solution by combining both greedy heuristics and randomization in a way that allows to construct a diverse number of good quality solutions. The post-processing attempts to improve the solution obtained in the first phase by means of local search.

Our GRASP construction phase has two stages. First a partial solution with p territories is built by using a GRASP greedy function that considers the objective function only. Then the remaining units are assigned to the territories by using a greedy function that incorporates some penalty terms associated to the violation of the balancing constraints. The post-processing phase consists of a local search scheme in which a node is selected to be moved from one territory to another. Both phases are described next.

Construction Phase: In this phase, the routing constraint is relaxed. After the solution is built, this constraint is taken into account in the local search phase. We attempt to build p territories one at a time. We first select one node to be the seed and then construct the territory by adding nodes (using a greedy function based only on a dispersion term) until a closing criteria is met. To start the next territory, we then select the minmax node, that is, the node whose minimum distance with respect to the territories constructed is the largest, and start adding nodes again until the same closing criteria is met. This is done until p territories are formed.

Depending on how tight or loose this user-defined “closing” criteria is, there may be still many unassigned nodes at the end of this step. The second step consists of assigning the remaining nodes by following an Adaptive Memory Programming scheme. In this step a merit function consisting of the sum of the original function, the violation of the balancing constraints, and a frequency-based memory term, is used.

Thus, for the first stage of the construction phase, let X_k be the partial territory being formed, and let N be the set of nodes adjacent to X_k, that is, the set of nodes that do not belong to X_k but that share an edge with a node in X_k. N is called the candidate list. For every $v \in N$ we evaluate the greedy function $\phi(v) = \max\{f_k(X_k), \max_{j \in X_k}\{t_{vj}\}\}$, where $f_k(X_k) = \max_{i,j \in X_k}\{t_{ij}^{X_k}\}$ is the contribution to the objective function of the k-th territory. The idea behind GRASP is to construct a restricted candidate list (RCL) containing the best moves. To do this a quality threshold parameter α is defined and the RCL is formed by those elements in N such that their corresponding greedy function evalution falls within α percent of the best move. That is RCL $= \{v \in N \ : \phi(v) \leq \Phi^{\min} + \alpha(\Phi^{\max} - \Phi^{\min})\}$, where $\alpha \in [0,1]$, $\Phi^{\min} = \min_{v \in N}\{\phi(v)\}$ and $\Phi^{\max} = \max_{v \in N}\{\phi(v)\}$. By defining the α this way, it is clear that a value of $\alpha = 0$ corresponds to a purely deterministic greedy approach, and $\alpha = 1$ corresponds to a purely randomized approach. We randomly choose one node from the RCL and we add it to V_k. In addition to dispersion, we also wish to have balanced territories according to both activities. As we are adding nodes to each territory, we are interested in not violating the upper bound of the balance constraints. Thus, if there is $a \in \{1,2\}$ such that $w^a(X_k) > \beta(1+\tau^a)\mu^a$ the territory is closed. β is a positive parameter that allows the user to control how early the territory must be closed. The motivation for introducing this parameter stems from the observation that allowing the territory to close earlier may give more flexibility for the remaining unassigned units to be assigned to different territories.

In Step 2, we use the greedy function

$$\varphi(v) = \lambda\phi(v) + (1-\lambda)G_k(v) + \vartheta(v),$$

where $\lambda \in [0,1]$, and $\phi(v)$ is the same greedy function used in the previous step,

$$G_k(v) = \sum_a (1/\mu^a) \max\left\{w^a(X_k \cup \{v\}) - (1+\tau^a)\mu^a, 0\right\}$$

represents the sum of relative infeasibilities with respect to the upper bound of the balance constraints and $\vartheta(v)$ is the average number of times node v ended up in the same territory than the rest of the nodes in the current territory, that is $\vartheta(v) = \sum_{i \in X_k} frec(i,v)/(|X_k| iter)$, where $frec(i,v)$ tallies the number of times nodes i and v have belonged to the same territory in the past $iter$ GRASP iterations. $iter$ is used as normalization factor. This represents the adaptive memory component.

Once the construction phase ends, we need to evaluate the routing costs. As stated before, solving the TSPM is equivalent to solving a TSP, so this is practicaly reduced to solving p TSPs, one for each territory. It is well-known that nowadays one can solve relatively large TSPs by branch-and-cut methods. In our case, the size of each of the individual TSPs to be solved for computing these costs is no more than 30 to 40 nodes. This implies one can use a branch-and-cut method for optimally solving each TSP in a relatively short amount of time. In this specific case, we use CONCORDE [1] to solve the corresponding TSP within each built territory. The connectivity requirement is kept during the entire procedure.

Post-processing Phase: The aim of the local search is to improve the objective function and at the same time to reduce the infeasibilities of the balance and the routing constraints as much as possible. The local search considers a neighborhood $N(S)$ of a partition S that consists of all possible movements of node i from its current territory $t(i)$ to the territory of another adjacent node j, $t(j)$, such that $(i,j) \in E$ and $t(i) \neq t(j)$.

We use a merit function with three terms. This merit function measures the objective function, the infeasibility of the budget constraint and the infeasibility of the balance constraints. This merit function is defined for a given partition $S = (X_1, \ldots, X_p)$ as follows:

$$\psi(S) = F(S) + \sigma H(S) + \gamma G(S) \tag{5}$$

where

$$F(S) = \max_k \left\{ \max_{ij \in X_k} \left\{ t_{ij}^{X_k} \right\} \right\}$$

is the original dispersion measure (diameter),

$$H(S) = \left(\frac{1}{C}\right) \max\left\{ \sum_k R(X_k) - C, 0 \right\} \tag{6}$$

is the relaxed budget constraint, and

$$G(S) = \sum_{k} \sum_{a \in A} g^a(X_k) \tag{7}$$

is the sum of all relative infeasibilities of the balance constraints, with

$$g^a(X_k) = \left(\frac{1}{\mu^a}\right) \max\left\{w^a(X_k) - (1+\tau^a)\mu^a, (1+\tau^a)\mu^a - w^a(X_k), 0\right\},$$

and σ and γ are penalty parameters to be dynamically updated as explained below.

Strategic oscillation: The parameters σ and γ in (5) are self-adjustable according to strategic oscillation [7]. When the budget constraint is violated, σ doubles its value and the same occurs with γ when the balance constraints are violated. When we have a feasible solution, both parameters reduce their values by half. With this strategy we can guide the search to a larger space by allowing infeasible moves. This technique has proven successfull in many combinatorial optimization problems, particularly in some territory design applications. For instance, Bozkaya, Erkut, and Laporte [2] make use of this idea for successfully handling some difficult constraints in a political districting problem. As it will be seen in the following section, this strategy gave very good results in our case as well.

4 Empirical Work

Our heuristic was coded in C++ and compiled with the GNU g++ version, under the Ubuntu Linux 9.10 OS in a computer with an Intel Processor(R) Core(TM)2 Quad CPU Q6600 of 2.40 GHz.

We used two types of instances: DU05 and DU10. Both types were taken from the database of Ríos-Mercado and Fernández [11]. DU05 instances have a balance deviation parameter $\tau = 0.05$ and DU10 have a $\tau = 0.10$. The budget limit C is given by the firm with a value of 20,000 for both types of instances. The size of these instances is of $n = 1000$ and $p = 40$. The β was set to 0.7, which was found to be the best setting in preliminary experiments.

4.1 Performance of Local Search

In this part of the experiment, we assess the contribution of the local search in its role of attempting to improve the quality of solutions and, more important, attempting to recover feasibility with respect to the solutions generated in the construction phase. To this end we run the procedure on 15 DU05 instances setting a budget limit of $C = 20{,}000$. The GRASP iteration limit was set at 1000.

Table 1 displays the results, where column 2 show the objective function value, and the third and fourth row show the average relative infeasibility with

Table 1. Improvement of local search

	Construction			Local Search		
Instance	f	RCI	BCI	f	RCI	BCI
1	181.29	0	9.3	213.7	0	0
2	178.17	0	12.4	204.6	1.9	0
3	168.90	0	5.1	186.4	0	0
4	184.71	0	7.5	238.3	0	0
5	207.96	0	7.7	176.9	0	0
6	180.21	0	8.1	191.4	0	0
7	210.16	0	8.8	206.8	0	0
8	185.76	0	9.6	191.7	3.6	0
9	172.81	0	11.1	228.4	0	0
10	203.86	0	10.3	197.5	0	0
11	166.99	0	8.5	188.6	0	0
12	181.33	0	7.1	175.4	0	0
13	182.97	0	7.2	181.2	0	0
14	174.58	0	11.6	202.8	0	0
15	200.43	0	9.5	223.4	6.2	0

respect to the budget (RCI) and balancing constraints (BCI) at the end of the construction phase, respectively. BCI is the sum of all the relative infeasibilities of each balancing constraints, which in turn is computed as the absolute violation of the balancing constraints divided by the corresponding average target size μ^a. For the budget constraint, RCI is computed as the absolute violation of the balancing constraint divided by the value of the upper bound C. RCI and BCI are given by $H(S)$ in (6) and $G(S)$ in (7), respectively. Columns 5-7 show the same statistics at the end of the local search.

It can be observed that the local search was very successfull on recovering feasibility (12 out of 15). In several instances the objective was also improved. This clearly indicates the excellent performance of the local search in reducing practically to zero the infeasibilities found at the end of the construction phase.

4.2 Assessment of Adaptive Memory

In order to evaluate the adaptive memory component, we compare our algorithm with (AM) and without (NAM) adaptive memory in both set of instances. In the following tables we are showing the relative improvement (RI) of AM over NAM, computed as

$$\text{RI} = 100 \times \frac{f_{NAM} - f_{AM}}{f_{NAM}}.$$

A negative (positive) value indicates a decrement (increment) in the objective.

Table 2. Evaluation of adaptive memory on DU05 instances

	NAM			AM			
Instance	*f*	RCI	BCI	*f*	RCI	BCI	RI (%)
1	213.71	0	0	178.97	0	0	16.2
2	204.64	1.9	0	249.58	0	0	-21.9
3	186.45	0	0	197.74	0	0	-6.0
4	238.36	0	0	253.95	0	0	-6.5
5	176.98	0	0	191.93	0	0	-8.4
6	191.43	0	0	188.31	0	0	1.6
7	206.85	0	0	189.28	0	0	8.4
8	191.70	3.6	0	179.45	2.8	0	6.3
9	228.42	0	0	220.76	2.3	0	3.3
10	197.57	0	0	204.57	0	0	-3.5
11	188.61	0	0	215.12	1.6	0	-14.0
12	175.41	0	0	176.20	0	0	-0.4
13	181.23	0	0	196.56	0	0	-8.4
14	202.85	0	0	187.51	0	0	7.5
15	223.42	6.2	0	225.37	6.2	0	-0.8
Average		0.8	0		0.9	0	0.29

Table 3. Evaluation of adaptive memory on DU10 instances

	NAM			AM			
Instance	*f*	RCI	BCI	*f*	RCI	BCI	RI (%)
1	212.63	0	0	190.39	0	0	10.4
2	221.88	3.0	0	193.58	0	0	12.7
3	188.37	0	0	242.14	0	0	-28.5
4	242.58	0.6	0	200.62	0	0	17.2
5	182.15	0	0	192.21	0	0	-5.5
6	227.27	0	0	240.25	0	0	-5.7
7	177.41	0	0	188.52	0	0	-6.2
8	191.59	0	0	190.43	0	0	0.6
9	188.08	0	0	196.87	0	0	-4.6
10	185.89	0	0	193.90	0	0	-4.3
11	165.82	0	0	192.55	0	0	-16.1
12	190.82	0	0	196.49	0	0	-2.9
13	196.94	0	0	195.65	0	0	0.6
14	175.59	0	0	191.76	0	0	-9.2
15	169.06	0	0	171.81	0	0	-1.6
Average		0.24	0		0	0	-5.86

Table 2 displays the results for DU05 instances. In this type of instances the results between the two strategies are very similar and it is not clear AM provides an advantage. We can see that there are three infeasible solutions under NAM and four infeasible solutions under AM. When comparing only the feasible solutions, we can see that in average there is an improvement of 0.29% of AM over NAM in solution quality, which is not too large.

Results for DU10 are shown in Table 3. In this case, AM turns out to be successfull, particularly in terms of finding feasible solutions. We can see that, in terms of comparing the objective function value, NAM does slightly better than AM. However, AM was succesfull on recovering feasibility by finding all 15 out of 15 feasible solutions. Procedure NAM failed in this regard in 14% of the instances. It can be concluded that the use of AM can result in a valuable strategy towards better design in terms of feasibility.

4.3 Assessment of Strategic Oscillation

As we recall from Section 3, in the proposed strategic oscillation parameters σ and γ penalize two terms in the merit function (5) changing dynamically whenever a movement is made and certain conditions are met. In this section we assess the performance of this strategy by comparing the algorithm with (SO) and without (NSO) the strategic oscillation. Both use the adaptive memory component. For NSO, we fixed parameters $\sigma = 10$ and $\gamma = 10$ and run the algorithm to observe the effect of not having a dynamic oscillation.

Table 4. Evaluation of strategic oscillation on DU10 instances

	NSO			SO		
Instance	f	RCI	BCI	f	RCI	BCI
1	218.00	0	5.5	190.39	0	0
2	233.96	0	12.0	193.58	0	0
3	192.67	0	5.1	242.14	0	0
4	183.99	0	5.9	200.62	0	0
5	261.74	0	5.2	192.21	0	0
6	238.43	0	4.3	240.25	0	0
7	200.25	0	4.2	188.52	0	0
8	231.74	0	6.4	190.43	0	0
9	192.25	0	7.0	196.87	0	0
10	243.85	0	7.7	193.90	0	0
11	187.31	0	6.0	192.55	0	0
12	227.93	0	5.7	196.49	0	0
13	196.04	0	8.1	195.65	0	0
14	211.91	0	5.5	191.76	0	0
15	198.04	0	4.0	171.81	0	0

Table 5. Evaluation of strategic oscillation on DU05 instances

	NSO			SO		
Instance	f	RCI	BCI	f	RCI	BCI
1	225.76	0	7.7	178.97	0	0
2	265.12	0	9.9	249.58	0	0
3	244.91	0	5.8	197.74	0	0
4	218.49	0	8.8	253.95	0	0
5	184.34	0	7.2	191.93	0	0
6	244.39	0	7.5	188.31	0	0
7	248.76	0	8.9	189.28	0	0
8	230.26	0	10.5	179.45	2.8	0
9	199.21	0	8.8	220.76	2.3	0
10	221.97	0	7.9	204.57	0	0
11	205.56	0	6.4	215.12	1.6	0
12	219.67	0	6.1	176.20	0	0
13	203.67	0	6.0	196.56	0	0
14	241.62	0	9.7	187.51	0	0
15	243.46	0	11.4	225.37	6.2	0

Tables 4 and 5 display the comparison for data sets DU10 and DU05, respectively. The information is similar to the one presented in the previous tables. In Table 4 we see that all the solutions reported by NSO satisfy the routing budget constraint, but not the balance constraints. The average relative infeasibilities are relatively high. This means this strategy struggled on trying to recover feasibility. Solutions obtained under SO are all feasible. The objective function values are better under SO as well. In Table 5 a similar behavior is observed. In this case, NSO could not obtain any feasible solution, and SO was successfull in finding feasible solutions in 11 out of 15 instances.

We can conclude directly that the strategic oscillation had a very positive impact leading the algorithm to feasible solutions and improving its solution quality. This clearly shows the excellent performance of the strategic oscillation.

5 Conclusions

We introduced a commercial Territory Design Problem with a routing budget constraint. This is, to the best of our knowledge, the first work to address both design and routing decisions within commercial territory design. To solve this problem we proposed a GRASP with some advance features such as adaptive memory and strategic oscillation.

The adaptive memory component was introduced during the construction phase as a diversification mechanism. The strategic oscillation was implemented within the local search to allow more flexibility in the search trajectory. The incorporation of these two into the procedure helped not only improve the quality of the solutions but to recover feasibility for almost all of them. Adaptive memory helped in terms of finding feasible solutions particularly for the DU10 instances. For the DU05 instances, the use of adaptive memory did not provide a significant advantage. In contrast, it was observed that the use of strategic oscillation was very successfull on both obtaining feasible designs, and improving solution quality.

Acknowledgments. This work was improved thanks to the remarks by three anonymous referees. This work was supported by grants SEP-CONACYT 48499-Y, from CONACYT, and CA1478–07 and CE012–09, from UANL-PAICYT.

References

1. Applegate, D., Bixby, R.E., Chvátal, V., Cook, W.: TSP Cuts Which Do Not Conform to the Template Paradigm. In: Jünger, M., Naddef, D. (eds.) Computational Combinatorial Optimization. LNCS, vol. 2241, pp. 261–304. Springer, Heidelberg (2001)
2. Bozkaya, B., Erkut, E., Laporte, G.: A tabu search heuristic and adaptive memory procedure for political districting. European Journal of Operational Research 144(1), 12–26 (2003)
3. Caro, F., Shirabe, T., Guignard, M., Weintraub, A.: School redistricting: Embedding GIS tools with integer programming. Journal of the Operational Research Society 55(8), 836–849 (2004)
4. Duque, J.C., Ramos, R., Suriñach, J.: Supervised regionalization methods: A survey. International Regional Science Review 30(3), 195–220 (2007)
5. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. Journal of Global Optimization 6(2), 109–133 (1995)
6. Fernández, E.A., Kalcsics, J., Nickel, S., Ríos-Mercado, R.Z.: A novel maximum dispersion territory design model arising in the implementation of the WEEE-directive. Journal of the Operational Research Society 61(3), 503–514 (2010)
7. Glover, F., Hao, J.K.: The case for strategic oscillation. Annals of Operations Research 183(1), 163–173 (2011)
8. Gutin, G., Punnen, A.P. (eds.): The Traveling Salesman Problem and its Variations. Springer, New York (2002)
9. Haugland, D., Ho, S.C., Laporte, G.: Designing delivery district for the vehicle routing problem with stochastic demands. European Journal of Operational Research 180(3), 997–1010 (2005)
10. Kalcsics, J., Nickel, S., Schröder, M.: Towards a unified territory design approach: Applications, algorithms, and GIS integration. TOP 13(1), 1–56 (2005)
11. Ríos-Mercado, R.Z., Fernández, E.A.: A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research 36(3), 755–776 (2009)
12. Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., Cabrera-Ríos, M.: New models for commercial territory design. Networks and Spatial Economics (Forthcoming, 2011) doi: 10.1007/s11067-010-9151-6
13. Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., González-Velarde, J.L.: GRASP strategies for a bi-objective commercial territory design problem. Journal of Heuristics (Forthcoming, 2011), doi: 10.1007/s10732-011-9160-8

Hybrid Intelligent Speed Control of Induction Machines Using Direct Torque Control

Fernando David Ramirez Figueroa and Alfredo Victor Mantilla Caeiros

Tecnologico de Monterrey, Campus Ciudad de Mexico,
Calle del Puente 222, Colonia Ejidos de Huipulco,
Tlalpan 14380, Distrito Federal, Mexico
{a00466442,amantill}@itesm.mx

Abstract. This paper presents a novel hybrid adaptive fuzzy controller for the regulation of speed on induction machines with direct torque control. The controller is based on a fuzzy system and PID control with decoupled gains. Genetic programming techniques are used for offline optimizations of the normalization constants of fuzzy membership function ranges. Fuzzy cluster means is introduced for online optimization on the limits of triangular fuzzy membership functions. Finally simulations in LabVIEW are presented validating the response of the controller with and without load on the machine; results and conclusions are discussed.

Keywords: Genetic Algorithms, Genetic Programing, Intelligent Control, Fuzzy Logic, Fuzzy C-means, FCM, Hybrid Intelligent Systems, DTC, Induction Machine.

1 Introduction

Induction machines are among the most widely used of all the electric motors. They are simple to build, rugged, and offer a reasonable asynchronous performance. Their torque-speed curve is controllable, they have a stable operation under load, and their efficiency is almost always satisfactory. Drives with high performance that control instantaneous electromagnetic torque (EMT) for induction motors (IM), have been in use for several decades. Direct torque control (DTC) was developed in the 1980's by Depenbrok [5], Takahashi and Noguchi [18].

Control techniques based on artificial intelligence have been very popular since the decade of the 1990's given that most of these techniques do not require complex mathematical models to be designed. Fuzzy controllers developed by Mamdani [12] and Takagi-Sugeno [17] are among the most popular systems. Other approaches consist of the use of Artificial Neural Networks [3, 14] to replace classical control regulators. Since the design of these controllers does not consider an optimal response and is normally based on human knowledge, genetic algorithms can be used to optimize response [7,10]. Numerous contributions have been made to DTC aided by Intelligent Control such as those presented in [4], [8], [9] and [13] among many others.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 319–330, 2011.

Improvements proposed in [8] and [9] are made to the DTC control scheme by using AI techniques instead of the conventional DTC algorithm. The work made in [13] proposes an adaptive law based on fuzzy logic which adjusts the gain on the model and estimates speed.

In this paper a novel speed control for induction motors using direct torque control and intelligent control techniques based on fuzzy logic, evolutionary algorithms, and fuzzy cluster means is presented. The performance of the speed hybrid controller increases as different intelligent control techniques are combined.

The purpose of this work is to apply artificial intelligence techniques not previously used to motor control. Conventional DTC is used to control torque and flux in the machine and indirectly regulate speed. The Controller is based on fuzzy Mamdani systems and PID control. A non-interactive (decoupled gains) version of PID is presented and implemented; the fuzzy controller is initially designed based on human knowledge. Later, this controller is optimized using genetic programing; finally Fuzzy Cluster Means is used to enhance the online performance.

This paper is organized as follows: section 1 presents an introduction, section 2 describes the theory of the proposed controller, section 3 shows simulations, section 4 presents and discusses tests and results and finally section 5 describe conclusions.

2 Intelligent Speed Control of Induction Machines Using DTC

A technique to indirectly regulate the speed of IM is by regulating torque (diagram shown in Figure 1). DTC decouples the control of torque and flux in an induction machine, making it very similar to the control of a direct current machine [19]. The speed controller is highlighted with a block with the broadest line; it will regulate the electromagnetic torque reference of the DTC loop, hence, indirectly regulating the speed.

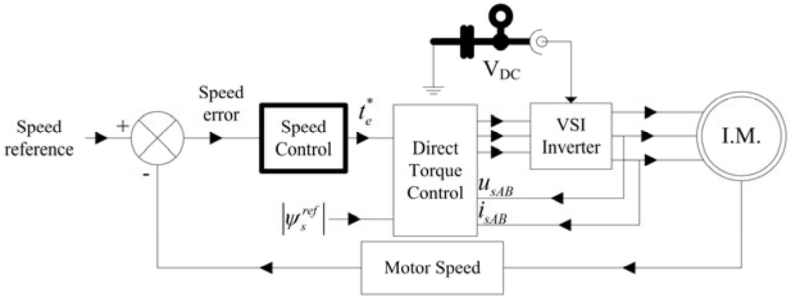


Fig. 1. Diagram of DTC for IM with speed loop using Intelligent Control

2.1 Basics of Direct Torque Control

In a DTC drive, flux linkage and electromagnetic torque are controlled directly and independently by selecting the optimum mode for the inverter. This selection is made to maintain flux and EMT errors within their respective hysteresis bands. This selection is also made to obtain a fast torque response, low inverter switching frequency, and low harmonics [5, 18]. The stator flux and EMT are restricted with their respective hysteresis bands, two and three levels respectively. The outputs of the comparators are used by the inverter switching table (Table 1), which also uses information of the sector of the stator flux space vector. In Table 1 the arrows indicate the increase or decrease of EMT or Flux, depending on the selected vector for action.

Table 1. Inverter switching table and their relation to flux and electromagnetic torque

	V_1	V_2	V_3	V_4	V_5	V_6	V_0 & V_7
$\lvert\psi\rvert$	↑↑	↑	↓	↓↓	↓	↑	---
EMT	↓	↑	↑	↓	↓↓	↓↓	↓

2.2 Mathematical Model of Induction Machine

The model of the induction machine is a dynamic vector model in a two axis DQ stationary reference frame, for a symmetrical squirrel cage induction machine. It is considered to be balanced and does not include the magnetic circuit model [15]. Here stator voltages are (1), rotor voltages (2). Flux linkages for stator and rotor are (3) and (4). EMT is described by (6) and finally stator, rotor and slip rotational speeds are related by (5).

$$v_{sDQ} = r_s i_{sDQ} + \frac{d\psi_{sDQ}}{dt} \quad (1) \qquad v_{rDQ} = 0 = r_r i_{rDQ} + \frac{d\psi_{rDQ}}{dt} - \omega_r \psi_{rQD} \quad (2)$$

$$\psi_{sDQ} = L_s i_{sDQ} + L_m i_{rDQ} \quad (3) \qquad \psi_{rDQ} = L_r i_{rDQ} + L_m i_{sDQ} \quad (4)$$

$$\omega_s = \omega_{slip} + \omega_r \quad (5) \qquad T_{em} = 3P/2J\left(\psi_{sD} i_{sD} - \psi_{sQ} i_{sQ}\right) \quad (6)$$

2.3 Adaptable Genetically Enhanced Fuzzy PID Controller

This controller is based on a non-interactive (decoupled gains) improved version of the classical PID control law; it will considerably increase the performance of the original version [1]. The control law is implemented using (7) and the signal is sent to the plant with (8).

$$\dot{u} = k_i e + k_p \dot{e} + k_d \ddot{e} \quad (7) \qquad u = \dot{u} + \Delta u \quad (8)$$

Each of the P, I, D gains are based in the Mamdani fuzzy system [12], shown in figure 2. The inputs and output of the controller are normalized to the [-2, 2] region and the outputs are scaled back to the operation range of the machine. Each fuzzy system (or gain) has three inputs, one output, triangular membership functions and three rules, generating a total of nine rules. The rules are shown in table 2: the general form of the rule is in (9).

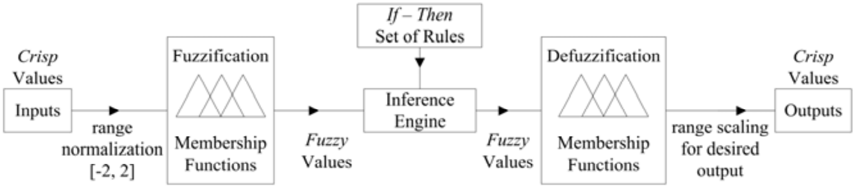


Fig. 2. Mamdani fuzzy inference system used for each of the PID gains in the control law

$$IF\ input\ is\ Inp_{MF}\ THEN\ output\ is\ Out_{MF}. \tag{9}$$

Where: *input* is a crisp value. *output*: is the crisp regulated value of EMT. Inp_{MF} or Out_{MF}: can be any of the membership functions (MFs) used in the fuzzification or defuzzification process. Once the response of each gain is calculated, the output is summed as in (7) and the control law sent to the plant is calculated with (8).

Table 2. Inputs, outputs and fuzzy rules of fuzzy PID controllers

Inputs	**Outputs for the three different gains**		
	Positive	**Zero**	**Negative**
Negative	Positive	Negative	Positive
Zero	Zero	Zero	Zero
Positive	Negative	Positive	Negative

$$Fitness = \frac{\sum\left(Speed_{ref} - Speed_{real}\right)^2}{\#Samples} \tag{10}$$ $$\#Samples = \frac{TotalSimTime}{StepSimTime} \tag{11}$$

The fuzzy PID is genetically enhanced using GP in the input and output normalization constants... The fitness is measured with the mean squared error calculated with (10), which is the reference speed minus the real speed squared, all of these samples summed and divided by the number of samples. The number of samples is the inverse of the step simulation time multiplied by the total simulation time (11).

2.3.1 Genetic Programming as Optimization Techniques

Genetic Programing -GP- creates autonomous programs that evolve their structure; this technique provides a way for searching for the best or *fittest* program to solve a problem. GP genetically breeds a population of programs using Darwin's natural selection and biological inspired operators. These programs are represented by trees. In GP the tree representation eliminates the problem of fixed size chromosomes making the search in a more organized manner [2, 10]. It also makes the search more organized, thus, being able to solve more complex problems than genetic algorithms.

Genetic Programming Algorithm Steps

GP is very similar to Genetic Algorithms because it contains biologically inspired operators. However, the differences cover some of the weaknesses of original GAs, as will be further described. The stages of the algorithm are [10]:

- *1. Initialization*: Elements such as number of generations, population size, probability of crossing, and mutating individuals, are created and initialized.
- *2. Selection*: Selection is made by measuring the performance of individuals aiming to maximize the performance of the population. The performance of individuals is measured using a *Fitness Function*. This function is a certain task where each individual is evaluated [16]. The tournament selection method is used, whereby a few randomly selected individuals compete in several tournaments and the winner of each tournament is selected for crossover.
- *3. Crossover*: This operation mates individuals by combining segments of chromosomes. Branches on trees are interchanged, diversifying the size of individuals and generating new ones.
- *4. Mutation*: When the mutation operation is performed, one of the branches of the individual is mutated, generating a new form in the branch and a new individual.
- *5. End conditions*: The algorithm will cease to operate if conditions, such as iteration number or a certain fitness value, are fulfilled. In the contrary case, the algorithm returns to *selection* to continue with the evolutionary process.

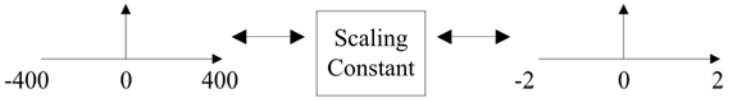


Fig. 3. Enhancements to the Fuzzy PID Controller

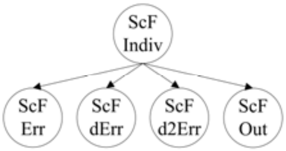


Fig. 4. Scaling factor optimization individual tree form

Normalization Factors Optimization

Normalization factors are very important since they dictate what portion of the decision table is used. They change membership functions uniformly over domains thus, changing the controllers gain over the whole domain uniformly. The scaling factors (Figure 3) are codified using 8 bit unsigned integers and later 16 bit unsigned integers to make the range of search wider.

The optimization process for scaling factors is done using a constant for each gain, along with different sizes for the number representation. The individual represented by a tree is shown in Figure 4. It is basically a tree with four branches; each one represents a scaling factor for the three inputs and one output.

2.3.2 Adjusting Membership Function for Optimal Response

The form of the membership functions is optimized online. Using the FCM algorithm the controller is able to adjust the form of its MFs online and with this information the limits of the triangular MFs are adjusted.

Fuzzy Cluster Means

Clustering methods split a set of N elements $X=\{x_1,\ldots,x_n\}$ into c groups or clusters $c=\{\mu^1,\ldots\mu^n\}$. Fuzzy set theory provides a natural way to describe clustering methods in a more realistic form using FCM. The FCM algorithm is described as follows: Fuzzy partition matrices M, for c classes and N data points are defined by three conditions $M\in\{U\in V_{cN}|(12)\}$:

$$\begin{gathered}\forall\ 1\le i\le c\quad \mu_{ik}\in[0,1],\quad 1\le k\le N;\quad \sum_{k=1}^{c}\mu_{ik}=1\quad \forall\quad 1\le k\le N\\ \forall\quad 1\le i\le c\quad 0<\sum_{K=1}^{c}\mu_{ik}<N\end{gathered}\tag{12}$$

Algorithm 1. Fuzzy Cluster Means algorithm

```
1. Fix $x$ and $m$, set p = 0 and initialize $U^{(0)}$.
2. Calculate fuzzy centers for each cluster $V^{(p)}$. using (16)
3. Update fuzzy partition matrix $U^{(p)}$ using (15)
4. If $||U^{(p)}-U^{(p-1)}||<\in$ then $j=+1$ and return to step 2.
```

FCM (algorithm 1) will maximize the distance between the centers of the clusters and minimize the distances of the elements of a same cluster. The FCM criteria function is shown in (13). d_{ik} is the inner product norm -distance- (14). A is a positive definite matrix and m the weighted exponent: $m\in[1,\infty)$. By assigning values to m and c and defining the working sets, (U,V) can be a global minimum of $J_m(U,V)$ if (15) and (16) are fulfilled [16]. Parameter m determines the fuzziness of the clusters; if $m = 1$ the algorithm becomes the crisp k-means version, and $m = \infty$ the algorithm is as fuzzy as possible, usually $m = 2$ [16].

$$J_m=(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}d_{ik}^{2}\tag{13}$$

$$d_{ik}^2 = \left\| x_k - v_i \right\|_A^2 \tag{14}$$

$$\forall\ 1 \le i \le c \quad 1 \le k \le N \qquad u_{ik} = 1 \Big/ \sum_{j=1}^{c} \left(\left\| x_k - v_i \right\| \big/ \left\| x_k - v_j \right\| \right)^{2/(m-1)} \tag{15}$$

$$\forall\ 1 \le i \le c \quad v_j = \sum_{k=1}^{N} u_{ik}^m x_k \Big/ \sum_{k=1}^{N} u_{ik}^m \tag{16}$$

The response of the controller can then be optimized depending on the input response and for that a simple algorithm is proposed and explained in Algorithm 2. Its execution can be graphically appreciated in figure 5. The form of the input MFs is varied only in their centers, so the internal action of the controller is not changed and erroneous response is induced. Nevertheless, adjusting the form of the output MFs in their entire limits will have a positive impact, since it will take action on the crisp output response of the controller.

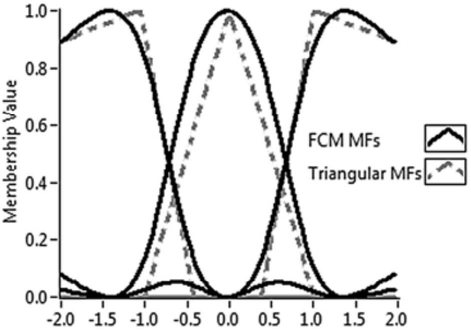


Fig. 5. Membership functions for the Adaptable Fuzzy PID Controller

Algorithm 2. Algorithm for the adaption of membership functions for optimization response

1. Generate a set of data in the range of action of the membership functions
2. Eliminate data from the initial set around a certain point
3.Adjust the form of the triangular membership functions using FCM

3 Simulation of the System

The implementation is made in LabVIEW [11] with the aid of the Intelligent Control Toolkit for LabVIEW [16]. For the PID controller the pre-optimization values are set as follows: the inputs are normalized by dividing them by 377, the top speed of the rotor [rad/s] and the output scaled back with a constant of 200. The values for these constants are set empirically based upon knowledge from the user; the simulation program is shown in Figure 6. The program that makes the optimizations using GP is shown in figure 7. The fitness function used is a simulation of the induction motor model with the DTC control scheme and the fuzzy PID loop controlling the speed, similar to the program in figure 6.

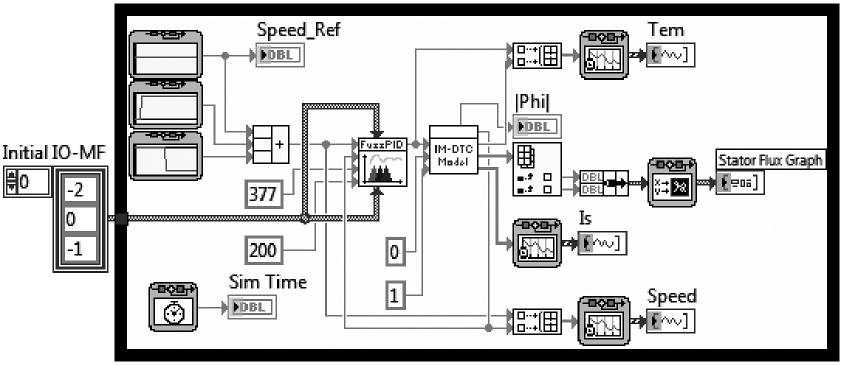


Fig. 6. DTC and IM loop with Fuzzy PID Controller in LabVIEW

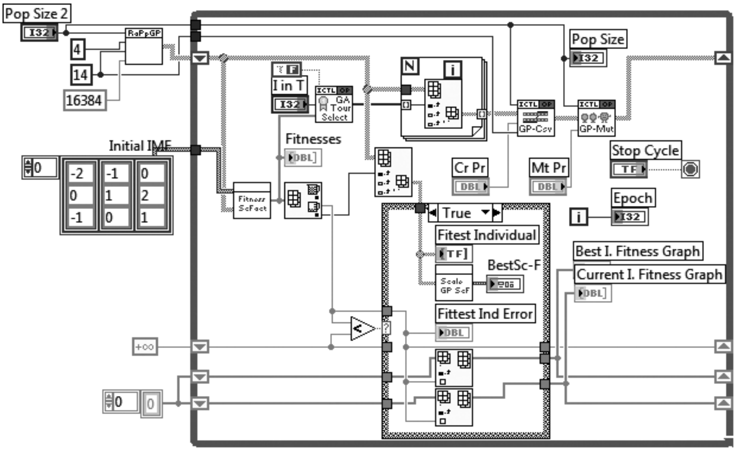


Fig. 7. of Genetic Programing block diagram system made in LabVIEW

4 Results

Parameters consulted in [15] for the induction motor model used in the simulations are shown in Table 4. Simulations have different speed references, as it is shown in Table 3, for a total of 10 s of simulation time. The solver used is Euler with a fixed step time of 50 μs. The GP optimizations had a fitness function with a period of time of 10 seconds, a step time of 0.2 ms and the Euler solver. Results are compared against a conventional PID with proportional gain of 1, integral gain of 10000 and derivative gain of 10000.

When the fuzzy controller is executed with the initial values prior to optimization the fitness value is 24674. Results of the GP are shown in Table 5; the fittest individual found had a fitness value of 17119.8. Figure 9 shows the results of this individual compared to a classical PID for the test presented in Table 3. The optimized controller (IC) takes the machine to the desired reference faster and more accurately than the classic PID (C-PID); thus showing that the response of the optimized controller is much better. These results show that GP is able to successfully optimize the overall response of the controller.

Table 3. Speed references test

Time Range [s]	Speed [rad/s]
[0, 1]	50
[1, 5]	300
[5, 10]	100

Table 4. Parameters for motor simulations where *p:* parameter and *v:* value

P	V	P	V	P	V	P	V
Rs	0.435 Ω	*Lm*	0.06931 H	σ	0.05531	*f*	60 Hz
Rr	0.816 Ω	*Ls, Lr*	0.07131 H	*Tm*	1 Nm	*Poles*	2

Table 5. Scaling factors optimization results using GP. *# Bits:* Bits used to represent the search space. *Pop Size:* Size of the population. *I in T:* Individuals used in tournament selection. *M Pr:* Probability of mutation. *Epochs:* Generations executed. *Fitness:* Fitness of the best individual found during that execution. *ScF Err:* Scaling Factor for the error. *ScF dErr:* Scaling Factor for the first derivate of the error. *ScF d2Err:* Scaling Factor for the second derivate of the error. ScF Out: Scaling Factor for the output. *Cross Probability* is 0.9 for all the executions.

# Bits	Pop Size	I in T	M Pr	Epochs	Fitness	ScF Err	ScF dErr	ScF d2Err	ScF Out
10	12	3	0.02	147	21399.71	266	0.0022	424	988
10	12	3	0.035	352	20655.92	585	0.0020	0.0020	988
10	16	4	0.025	60	21399.7	195	0.0025	214	834
13	24	6	0.01	35	19261.2	7182	0.0004	0.0003	5168
13	***14***	***4***	***0.015***	***82***	***17119.8***	***512***	***84***	***0.0002***	***8144***
13	14	4	0.015	194	19247.7	6114	0.0003	0.0005	1822
16	24	6	0.01	206	19250.93	55378	3.3E-5	5.7E-4	54277

Once GP optimizations were executed the adaptable algorithm on the controller was allowed to operate and be tested. The adjusted membership functions are shown in Figure 8, the left shows the input MFs and the right the output MFs. The shapes of the input MFs are optimized with the information of the fuzzy centers coming from FCM, the beginning and ending limits of the zero cluster.

The limits of the output MFs are also optimized with the centers coming from FCM, the ending limit of negative cluster, beginning and ending of zero cluster, and beginning of the positive cluster. The results for the test presented in Table 3 are shown in Figure 9; as it can be seen the response of the controller is optimal, being able to set the speed of the motor to the desired reference and update the limits of its membership functions on the fly.

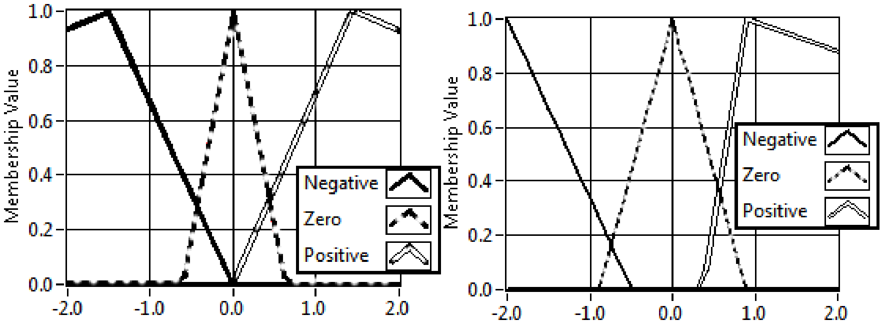


Fig. 8. Triangular membership functions adjusted with adaptable MF algorithm

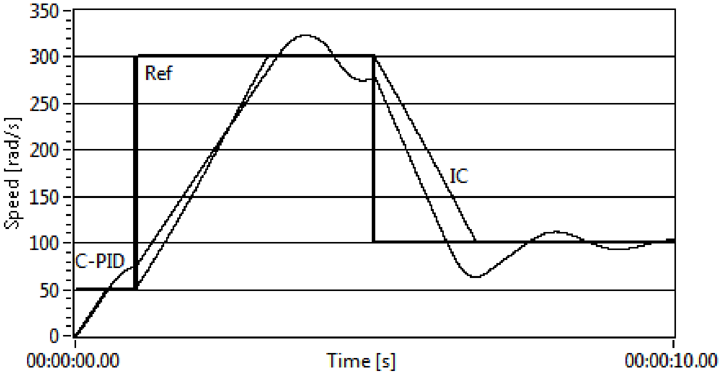


Fig. 9. Speed test for Adjustable fuzzy controller without load

Although the response of the controller has been optimized successfully under no load to the machine, it is interesting to analyze what would happen if load is introduced to the motor. For that reason a simple test is shown in Table 6; after the controller has reached a desired speed, different loads are introduced to the machine.

Table 6. Speed references test with load

Time Range [s]	Torque Load [Nm]
[4.2, 5.5]	50
[5.5, 9]	80

As can be appreciated in Figure 10, when the first load of 50Nm is introduced, the adjustable controller response does not change, while the response of the non-adjustable controller drops, but remains constant.

When the 80Nm load is set the response of all the controllers drops; however, after some moments the adjustable controller is able to compensate and the response of the machine increases. The non-adjustable controller is not able to compensate and the response goes down until the load is withdrawn. After the load is withdrawn both controllers compensate and correct the response. A small steady state error can be seen in the normal fuzzy controller, while the optimized controller does not show the same steady state error. Finally, the response of the classic PID (C-PID) is of the chart limits, with very high oscillations and not being capable of compensating the load.

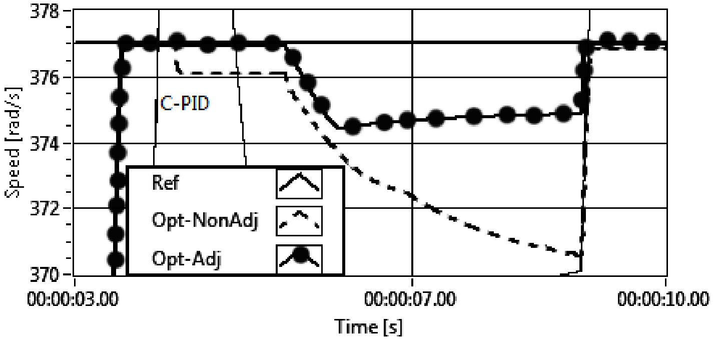


Fig. 10. Speed test for Adjustable fuzzy controller with load

5 Conclusions

This work presents a novel approach on intelligent control for the speed control on induction machines validated through simulations. Most of these applications require complex computational capabilities, which are getting easier to obtain at lower costs. The proposed PID decoupled topology was validated and can be easily transferable to an embedded system. Difference gives more information because they represent a reason of change. Digital systems saturate integrators and differences can be easily computed with subtractions.

The optimizations based on GP proved to effectively optimize the fuzzy controllers. GP can deal with complex forms of optimizations; these optimizations have to be carried offline due to the nature of evolutionary algorithms and their delayed response due to large search spaces.

The online optimization proposed proved to correctly compensate unknown uncertainties introduced to the system and further improve the response of the controllers. Furthermore, this optimization can be executed online without requiring great computational load, as it is executed with a 1 ms period, much greater to the smaller -lower than 25 μs- required executing DTC [6].

References

1. Astrom, K., Hagglund, T.: PID Controllers: Theory, Design and Tuning. Instrument Society of America, USA (1995)
2. Bourmistrova, A., Khantis, S.: Control system design optimisation via genetic programming. In: IEEE Congress on Evolutionary Computation, CEC 2007, Singapore, pp. 1993–2000 (2007)
3. Cheng-Zhi, C., Guang-Hua, W., Qi-Dong, Z., Xin, W.: Optimization Design of Fuzzy Neural Network Controller in Direct Torque Control System. In: Third International Conference on Machine Learning and Cybemetics, Shanghai, vol. 1, pp. 378–382 (2004)
4. Stephen, C.: Developing commercial applications of intelligent control. IEEE Control Systems Magazine 17(2), 94–100 (1997)
5. Depenbrock, M.: Direkte Selbtregelung (DSR) für hochdynamische Drehfeldantribe mit Stromrichterschaltung. ETZ A 7, 211–218 (1985)
6. Dufoo, S., Pacas, M.: Predictive Direct Torque Control of an Induction Machine with Unsymmetrical Rotor. In: IEEE International Conference on Industrial Technology, pp. 1851–1856 (2010)
7. Goldbergh, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Massachusetts (1989)
8. Grabowski, P.Z., Blaabjerg, F.: Direct Torque Neuro-Fuzzy Control of Induction Motor Drive. In: 23rd International Conference on Industrial Electronics, Control and Instrumentation, vol. 2, pp. 557–562 (1998)
9. Li, H.: Fuzzy DTC for Induction Motor with Optimized Command Stator Flux. In: 8th World Congress on Intelligent Control and Automation, Jinan, China, pp. 4958–4961 (2010)
10. Koza John, R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
11. LabVIEW, graphical programming language, `http://www.ni.com/labview`
12. Mamdani, E.H.: Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. Institution of Electrical Engineers, Control & Sciences 121(12), 1585–1588 (1974)
13. Sui, M., Zhang, K., Yang, J.: An Improved Sensorless DSVM-DTC of Induction Motor Based MRAFC. In: 7th World Congress on Intelligent Control and Automation, pp. 775–780 (2008)
14. Pedro., P.C., Rivas, J.J.R.: A Small Neural Network Structure Application in Speed Estimation of an Induction Motor Using Direct Torque Control. In: IEEE 32nd Annual Specialists Conference on Power Electronic, vol. 2, pp. 823–827 (2001)
15. Pedro, P.C., Javier, S.: Maquinas Electricas y Tecnicas Modernas de Control. Grupo, A. (ed.), Mexico (2008)
16. Pedro, P.-C., Fernando, D.: Ramirez-Figueroa. In: Intelligent Control Systems with LabVIEW. Springer, London (2009)
17. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15, 116–132 (1985)
18. Takahashi, I., Noguchi, T.: Quick torque response control of an induction motor using a new concept. IEEE J. Tech. Meeting on Rotating Machines, paper RM84-76, 61–70 (1984)
19. Peter, V.: Sensorless Vector and Direct Torque Control. Oxford University Press (2003)

A New Model of Modular Neural Networks with Fuzzy Granularity for Pattern Recognition and Its Optimization with Hierarchical Genetic Algorithms

Daniela Sánchez, Patricia Melin, and Oscar Castillo

Tijuana Institute of Technology
pmelin@tectijuana.mx

Abstract. In this paper we propose a new model of a Modular Neural Network (MNN) with fuzzy integration based on granular computing. The topology and parameters of the model are optimized with a Hierarchical Genetic Algorithm (HGA). The model was applied to the case of human recognition to illustrate its applicability. The proposed method is able to divide the data automatically into sub modules, to work with a percentage of images and select which images will be used for training. We considered, to test this method, the problem of human recognition based on ear, and we used a database with 77 persons (with 4 images each person for this task).

Keywords: Modular Neural Networks, Fuzzy Logic, Genetic Algorithms, Granular computing.

1 Introduction

To solve complex problems, a great diversity of techniques are required. These techniques can be divided into two categories: (1) traditional hard computing techniques (as operations research, system science/ engineering, expert systems), and (2) soft computing techniques (such as fuzzy logic (FL), neural networks (NN), and genetic algorithms (GA)). These techniques are complementary thus they may be used in combination. These systems are called hybrid intelligent systems.

Hybrid intelligent systems are computational systems that integrate different intelligent techniques. These systems are now being used to support complex problem solving and decision making in a wide variety of tasks. Hybrid intelligent systems allow the representation and manipulation of different types and forms of data and knowledge which may come from various sources [30].

Human problem solving involves the perception, abstraction, representation and understanding of real world problems, as well as their solutions, at different levels of granularity [11],[14],[25],[26],[27],[28]. The consideration of granularity is motivated by the practical needs for simplification, clarity, low cost, approximation, and tolerance of uncertainty [17],[20],[24],[25],[26]. However, granular computing has not been fully explored in its own right. It is time to extract the commonality from these diverse fields and to study systematically and formally the domain independent principles of granular computing in a unified and well-formulated framework [24].

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 331–342, 2011.

This paper is organized as follows. In Section 2 basic concepts are presented. In Section 3, the description of the proposed method is offered and also its tested using a benchmark database. The results obtained of testing the proposed method are explained in Section 4. Conclusions are presented in Section 5.

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 have a remarkable ability to derive meaning from complicated or imprecise data, and they can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest [16]. A neural network is said to be modular if the computation performed by the network can be decomposed into two or more modules (subsystems) that operate on distinct inputs without communicating with each other. The modular neural networks are comprised of modules, which can be categorized on the basis of both distinct structure and functionality which are integrated together via an integrating unit. With functional categorization, each module is a neural network which carries out a distinct identifiable subtask. Also, using this approach different types of learning algorithms can be combined in a smooth fashion [1].

2.2 Granular Computing

Granular computing is often defined as an umbrella term to cover any theories, methodologies, techniques, and tools that make use of granules in complex problem solving [23],[24]. Granular computing is a new term for the problem solving paradigm and may be viewed more on a philosophical rather than technical level [22],[23].

Granular Computing (GrC), as defined in the outline of the IEEE-GrC'2006 conference information, is a general computation theory for effectively using granules such as classes, clusters, subsets, groups and intervals to build an efficient computational model for complex applications with huge amounts of data, information and know-ledge. Though the label is relatively recent, the basic notions and principles of granular computing, though under different names, have appeared in many related fields, such as information hiding in programming, granularity in artificial intelligence, divide and conquer in theoretical computer science, interval computing, cluster analysis, fuzzy and rough set theories, neutrosophic computing, quotient space theory, belief functions, machine learning, databases, and many others. In the past few years, we have witnessed a renewed and fast growing interest in GrC.

Granular computing has begun to play important roles in bioinformatics, e-Business, security, machine learn-ing, data mining, high-performance computing and wireless mobile computing in terms of efficiency, effectiveness, robustness and uncertainty [2] ,[22].

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., an 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 [11],[12],[13].

Uncertainty in the primary memberships of a type-2 fuzzy set, $\tilde{A}$, consists of a bounded region that we call the "footprint of uncertainty" (FOU). Mathematically, it is the union of all primary membership functions [4],[5],[18].

A type-2 fuzzy set $\tilde{A}$, is characterized by the membership function (see expression 1):

$$\tilde{A}= \{((x,u), \mu_{\tilde{A}}\ (x,u)) \mid \forall x \in\ X, \forall u \in\ J_x \subseteq [0,1]\ \} \tag{1}$$

where x means the input variable, u means a type-1 membership function, J_x means an interval $\subseteq [0,1]$, and $\mu_{\tilde{A}}$ means a type-2 membership function. Another expression (2) for A is,

$$\tilde{A}= \int_{x\in X}\int_{u\in Jx} \mu_{\tilde{A}}\,(x,u)/\,(x,u) \qquad\qquad J_x \subseteq [0,1] \tag{2}$$

The distinction between type-1 and type-2 is associated with the nature of the membership functions, which is not important when forming the rules. The structure of the rules remains exactly the same in the type-2 case, but now some or all of the sets involved are type-2.

Consider a type-2 FLS having r inputs $x_1 \in X_1, \ldots, x_r \in X_r$ and one output $y \in Y$. As in the type-1 case, we can assume that there are M rules; but, in the type-2 case the lth rule has the form

$$R^1 : \text{IF } x_1 \text{ is } \tilde{A}^1{}_1 \text{ and} \ldots x_p \text{ is } \tilde{A}^1{}_p \text{, THEN y is } Y^1 \qquad\qquad 1=1,\ldots,M$$

This rule represents a type-2 fuzzy relation between the input space $X_1 \times \ldots \times X_r$, and the output space, Y, of the type-2 fuzzy system.

If we considered two fuzzy sets (type-2) named $\tilde{A}_1$ and $\tilde{A}_2$ their union is another type-2 fuzzy set, just as the union of type-1 fuzzy sets A_1 and A_2 is another type-1 fuzzy set. More formally, we have the following expression (3)

$$\tilde{A}1 \cup \tilde{A}2 = \int_{x \in X} \mu_{\tilde{A}1 \cup \tilde{A}2} (x)/\, x \qquad (3)$$

The intersection of $\tilde{A}_1$ and $\tilde{A}_2$ is another type-2 fuzzy set, just as the intersection of type-1 fuzzy sets A_1 and A_2 is another type-1 fuzzy set. More formally, we have the following expression (4)

$$\tilde{A}1 \cap \tilde{A}2 = \int_{x \in X} \mu_{\tilde{A}1 \cap \tilde{A}2} (x)/\, x \qquad (4)$$

The complement of set $\tilde{A}$ is another type-2 fuzzy set, just as the complement of type-1fuzzy set A is another type-1 fuzzy set. More formally we have the following expression (5)

$$\tilde{A}' = \int_{x} \mu_{\tilde{A}'1} (x)/\, x \qquad (5)$$

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 [3].

2.4 Genetic Algorithms

Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure, and apply recombination operators to these structures in such a way as to preserve critical information. Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to 'reproduce' than those chromosomes which are poorer solutions. The 'goodness' of a solution is typically defined with respect to the current population [21],[29].

GAs have proven to be a useful method for optimizing the membership functions of the fuzzy sets used by these fuzzy systems [19]. There are also other methods used in other works for optimization (of neural networks and fuzzy systems) such as particle swarm optimization (PSO), in these works [7],[8],[9] they used PSO for the optimization of artificial neural networks (ANN), in this paper we only used genetic algorithms for the optimization of modular neural networks (MNN).

3 General Architecture of the Proposed Method

The proposed method combines modular neural networks (MNN) and fuzzy logic as response integrators. In particular, it can be used for pattern recognition. This pro-posed method is able to use some data sets, for example to use "N" biometric measures to identify someone and the data of each biometric measure will be able to be divided into different numbers of sub modules. We can notice in Figure 1 the general architecture of the proposed method. For joining the different responses of each biometric measure fuzzy integration is used. The proposed method also performs the optimization of the modular neural networks (as number of layers, goal error, number of neurons, etc.) and the different parameters of the fuzzy integrator.

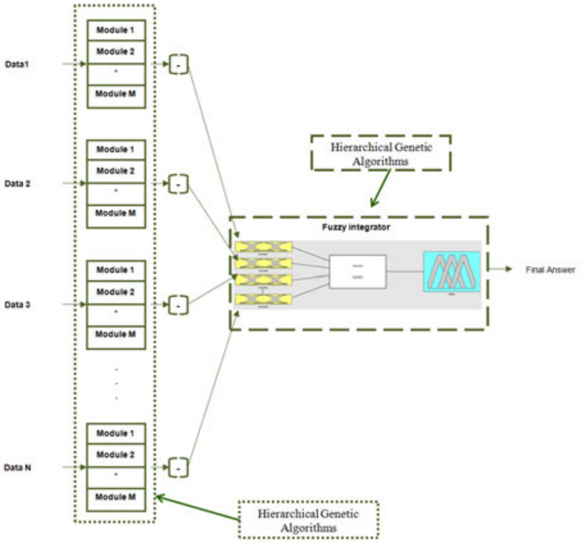


Fig. 1. The general architecture of proposed method

3.1 General Architecture of the Proposed Method for the Modular Neural Network

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. The number of sub modules can be established by a genetic algorithm, but in this moment the number is established randomly. We can notice in Figure 2 the architecture of the proposed method for the modular neural network.

This method also chooses randomly which images will be used for training, but first the percentage of images for training is established (at this moment that percentage is defined randomly).

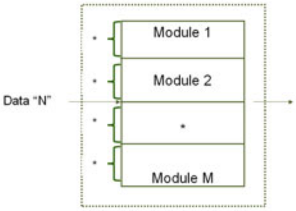


Fig. 2. The architecture of proposed method for the modular neural network

3.2 Description of the Genetic Algorithm for MNN Optimization

With the purpose of knowing the optimal number of modules and the percentage of data for training, it is proposed the use of a genetic algorithm that allows the optimization of these parameters and others as the number of hidden layers, number of neurons per hidden layer, error goal and learning algorithms per module.

Figure 3 shows the chromosome, which was proposed for optimization of the neural networks.

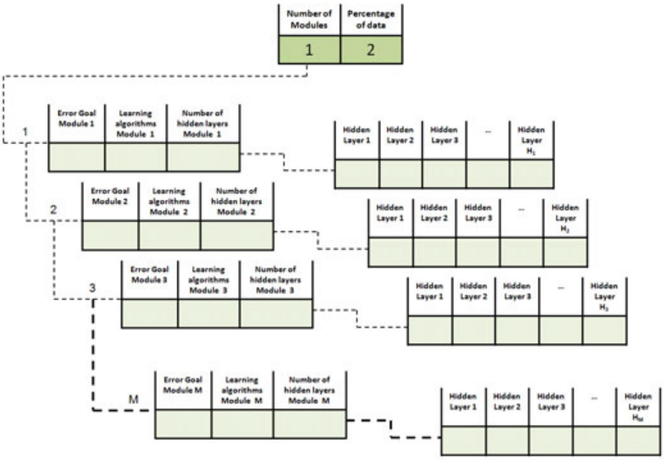


Fig. 3. The chromosome of genetic algorithm for the MNN

3.3 General Architecture of the Proposed Method for the Fuzzy Integrator

The number of inputs of the fuzzy integrator will depend for example on the number of biometric measures used for identification. Figure 4 shows an example of a fuzzy integrator. The proposed method for the fuzzy integrator consists in changing the number of membership functions, the type of membership functions (it is important to mention that this method allows to combine different type of membership functions, type of system (Mamdani or Sugeno) and rules. Of course, Type-2 fuzzy logic is considered in all cases.

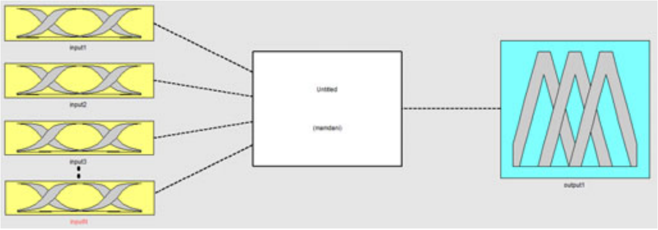


Fig. 4. Example of the fuzzy integrator

We can notice in Figure 5 an example of how this method allows a combination of different membership functions.

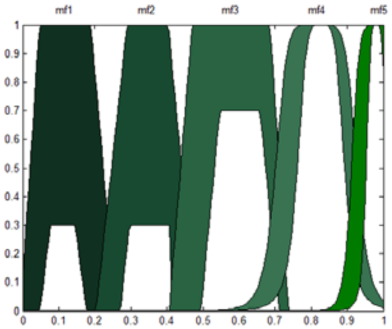


Fig. 5. Example of combination of membership functions

3.4 Description of the Genetic Algorithm for the Fuzzy Integrator

For the first test a Genetic Algorithm was proposed for the optimization of the fuzzy integrator, in this case the number of membership functions is optimized (1 to 3), parameters of these membership functions, and only generalized bell membership functions are used. Figure 6 shows an example of the chromosome a fuzzy integrator of Mamdani type.

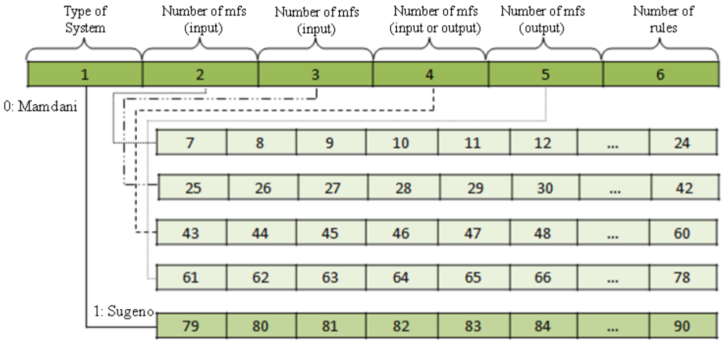


Fig. 6. The chromosome of genetic algorithm for the fuzzy integrator

3.5 Database of Ear

We used a database of the University of Science and Technology of Beijing [6]. The database consists of 77 people, which contains 4 images per person (one ear), the image dimensions are 300 x 400 pixels, the format is BMP. 3 images were used for training and 1 for testing.

Figure 7 shows an example of pre-processing applied to each image in the ear, it is a manual cut 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 3 parts of interest (helix, lobe and shell). In this case we worked in the optimization for the fuzzy system that it is used for integration in the modular neural network (MNN).

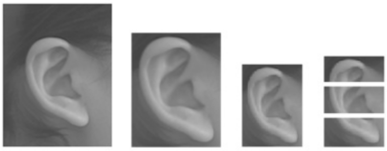


Fig. 7. Sample pre-processing done to the images of ear

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 1 Module with Fuzzy Integration

In this test we used for training the images 1, 2 and 3, and for testing image 4. For the integration of responses we used a fuzzy integrator, this fuzzy integrator was optimized with the genetic algorithm (the optimized parameters were already mentioned).

We performed a number of tests with the genetic algorithm adding noise and the results are shown in Figure 8. The best fuzzy system with noise is shown in Figure 9.

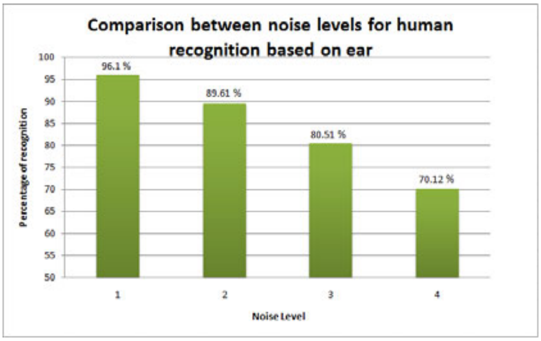


Fig. 8. Comparison among different noise levels

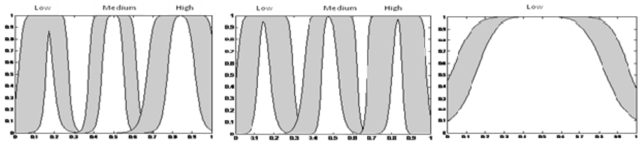


Fig. 9. Membership function of helix, shell, lobule

4.2 Results with 3 Modules

In this test we established that the number of modules is 3, the variables that were established randomly were the percentage of images used and the images which would be used for training. We can notice in Table 1 the best results for ear with 3 modules. For the integration of responses of this and the next test the winner takes all integration method was used.

Table 1. The best results for ear (with 3 modules)

Training	Images for training	Persons per module	Recognition Rate
1	1 and 4	Module # 1 (1 to 24) Module # 2 (25 to 65) Module # 3 (66 to 77)	66.88% (103/154)
2	1,3 and 4	Module # 1 (1 to 6) Module # 2 (7 to 14) Module # 3 (15 to 77)	67.53% (52/77)
3	2 and 4	Module # 1 (1 to 38) Module # 2 (39 to 70) Module # 3 (71 to 77)	77.92% (120/154)
4	1 and 3	Module # 1 (1 to 9) Module # 2 (10 to 44) Module # 3 (45 to 77)	83.11% (128/154)
5	3	Module # 1 (1 to 14) Module # 2 (15 to 50) Module # 3 (51 to 77)	53.67% (124/231)
6	2	Module # 1 (1 to 38) Module # 2 (39 to 61) Module # 3 (62 to 77)	54.97% (127/231)
7	2, 3 and 4	Module # 1 (1 to 40) Module # 2 (41 to 50) Module # 3 (51 to 77)	100% (77/77)
8	2 and 3	Module # 1 (1 to 23) Module # 2 (24 to 47) Module # 3 (48 to 77)	93.50% (144/154)

4.3 Results with Different Number of Modules

In this test the variables that were established randomly were the number of modules, the percentage of images and the images which would be used for training. We can notice in Table 2 the best results for ear with different number of modules.

In the second column, the images that were used for training are shown. In the third column, the number of modules used in each training are shown. In the fourth column, the number of persons per module is shown, and finally, in the fifth column the final achieved recognition rates are shown. We can notice that depending on the

Table 2. The best results for the ear (with different number of modules)

Training	Images for training	Number of modules	Persons per module	Recognition Rate
1	1,2 and 3	8	Module # 1 (1 to 2) Module # 2 (3 to 11) Module # 3 (12 to 25) Module # 4 (26 to 36) Module # 5 (37 to 43) Module # 6 (44 to 58) Module # 7 (59 to 62) Module # 8 (63 to 77)	100% (77/77)
2	2 and 3	5	Module # 1 (1 to 5) Module # 2 (6 to 13) Module # 3 (14 to 49) Module # 4 (50 to 52) Module # 5 (53 to 77)	89.61% (138/154)
3	2 and 4	3	Module # 1 (1 to 11) Module # 2 (12 to 51) Module # 3 (52 to 77)	81.16% (125/154)
4	2 and 4	6	Module # 1 (1 to 18) Module # 2 (19 to 30) Module # 3 (31 to 45) Module # 4 (46 to 51) Module # 5 (52 to 59) Module # 6 (60 to 77)	83.11% (128/154)
5	1	9	Module # 1 (1 to 13) Module # 2 (14 to 17) Module # 3 (18 to 19) Module # 4 (20 to 26) Module # 5 (27 to 41) Module # 6 (42 to 45) Module # 7 (46 to 53) Module # 8 (54 to 69) Module # 9 (70 to 77)	69.69% (161/231)
6	3 and 4	9	Module # 1 (1 to 6) Module # 2 (7 to 14) Module # 3 (15 to 18) Module # 4 (19 to 26) Module # 5 (27 to 42) Module # 6 (43 to 51) Module # 7 (52 to 58) Module # 8 (59 to 65) Module # 9 (66 to 77)	87.01% (134/154)

way that the data were distributed, the results change. In this test the best result corresponds to training #1 with a rate of recognition of 100%. In this training, 3 images were used (1,2 and 3) for training, and 8 modules for the MNN. Those data were obtained randomly with a function created by ourselves. In [10] the same database was used and with cross validation, obtained an average of recognition rate of 94.48%, but they used 2D wavelet analysis for preprocessing.

5 Conclusions

We performed different tests changing the number of modules, we can notice that when the MNN is used with one module we can obtain a rate of recognition of 100%, but it is important to realize that in previous work already done we already had obtained this rate with the same images for training.

We can notice that when our method changed the images for testing and the number of modules the results are affected. We are going to work with a genetic algorithm for the optimization the different parameters of the modular neural network as number of modules and the percentage of images for training. We will also use other benchmark databases for testing our method.

References

1. Azamm, F.: Biologically Inspired Modular Neural Networks, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (2000)
2. Bargiela, A., Pedrycz, W.: The roots of granular computing. In: Proc. IEEE Granular Computing Conf., p. 741 (2006)
3. Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications, pp. 29–43. Springer, Berlin (2008)
4. 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)
5. 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)
6. 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)
7. Garro, B.A., Sossa, H., Vazquez, R.A.: Design of Artificial Neural Networks using a Modified Particle Swarm Optimization Algorithm. In: International Joint Conference on Neural Networks (IJCNN 2009), Atlanta, GE, USA, June 14-19, pp. 938–945 (2009)
8. Garro, B.A., Sossa, H., Vázquez, R.A.: Design of Artificial Neural Networks Using Differential Evolution Algorithm. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part II. LNCS, vol. 6444, pp. 201–208. Springer, Heidelberg (2010)
9. Garro, B.A., Sossa, H., Vázquez, R.A.: Evolving Neural Networks: A Comparison Between Differential Evolution and Particle Swarm Optimization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part I. LNCS, vol. 6728, pp. 447–454. Springer, Heidelberg (2011)
10. Gutiérrez, L., Melin, P., López, M.: Modular Neural Network for Human Recognition From Ear Images Using Wavelets. In: Melin, P., Kacprzyk, J., Pedrycz, W. (eds.) Soft Computing for Recognition Based on Biometrics. SCI, vol. 312, pp. 121–135. Springer, Heidelberg (2010)

11. 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)
12. 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: Soft Computing for Hybrid Intelligent Systems, pp. 89–114 (2008)
13. 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)
14. Hobbs, J.R.: Granularity. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence, pp. 432–435 (1985)
15. Huang, J., Wechsler, H.: Eye Location Using Genetic Algorithm. Department of Computer Science, George Mason University, Washington, DC (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. Mendel, J.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
19. Nawa, N., Takeshi, F., Hashiyama, T., Uchikawa, Y.: A study on the discovery of relevant fuzzy rules using pseudo-bacterial genetic algorithm. Laboratory of Bio-Electronics, Department of Information Electronics, School of Engineering, Nagoya University, Japan (1999)
20. Pedrycz, W.: Granular Computing: An Emerging Paradigm. Physica-Verlag, Heidelberg (2001)
21. Whitley, D.: A genetic algorithm tutorial. Statistics and Computing 4, 65–85 (1994)
22. Yao, J.T.: A ten-year review of granular computing. In: Proceedings of the 3rd IEEE International Conference on Granular Computing (2007)
23. Yao, J.T.: Information granulation and granular relationships. In: Proceedings of 2005 IEEE Conference on Granular Computing, Beijing, China, pp. 326–329 (2005)
24. Yao, Y.Y.: Granular computing: basic issues and possible solutions. In: Proceedings of the 5th Joint Conferences on Information Sciences, New Jersey, USA, pp. 186–189 (2000)
25. Yao, Y.: A Partition Model of Granular Computing. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 232–253. Springer, Heidelberg (2004)
26. 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)
27. 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)
28. Zhang, L., Zhang, B.: The Quotient Space Theory of Problem Solving. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 11–15. Springer, Heidelberg (2003)
29. Zhang, L., Zhang, B.: Theory and Application of Problem Solving. Elsevier Science Publishers, North- Holland (1992)
30. Zhang, Z., Zhang, C.: An Agent-Based Hybrid Intelligent System for Financial Investment Planning. In: Ishizuka, M., Sattar, A. (eds.) PRICAI 2002. LNCS (LNAI), vol. 2417, pp. 355–364. Springer, Heidelberg (2002)

Crawling to Improve Multimodal Emotion Detection

Diego R. Cueva, Rafael A. M. Gonçalves,
Fábio Cozman, and Marcos R. Pereira-Barretto

Departamento de Engenharia Mecatrônica e Sistema Mecânicos
Escola Politécnica da Universidade de São Paulo (EPUSP)
Av. Prof. Melo Moraes 2231 – São Paulo – SP - Brazil
marcos.barretto@poli.usp.br

Abstract. This paper demonstrates multimodal fusion of emotion sensory data in realistic scenarios of relatively long human-machine interactions. Fusion, combining voice and facial expressions, has been enhanced with semantic information retrieved from Internet social networks, resulting in more accurate determination of the conveyed emotion.

Keywords: Affective Computing, Emotion Detection, Artificial Intelligence, Web Crawling, Sensor Fusion.

1 Introduction

On the ongoing evolution of computing, the development of friendly user experiences has presented restrained innovation since the dawn of the first fully graphical applications in the 70's. Although graphical improvements of such systems have been of great relevance, the lack of a paradigm change is hindering richer interaction between humans and machines. Without such a change, humans will still have to learn and adapt to the manner of operation of each machine, always strictly bounded by use cases established by the developer.

To emulate the human capacity of contextualization of a conversation and the flexible reaction to its semantic meaning would be the most natural way of interaction: if computers had the ability to process cues from the several linguistic and non-linguistic signs that permeate human behavior (like facial expressions, tone of voice and the affective context), they would be capable of comprehending in a more appropriate way the needs and issues of the user, bringing us closer to a genuinely human centered interaction.

Driven by these issues, several studies aim to develop computational algorithms to detect emotions, whether extracted from facial or vocal features. In general, these studies support themselves on the reputable and widely used facial classification developed by Ekman and Friesen [1], which distinguishes six basic blocks of emotional expressions: happiness, sadness, disgust, fear, anger and surprise.

In the last decade, with the vast expansion of opinion-oriented Internet databases (blogs, forums and social networks), some studies have also attempted to observe

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 343–350, 2011.

emotions in basic semantic elements of conversation, comparing them with common sense available on the web.

However, although the field in question is rapidly developing, the existing techniques for dealing with emotions are still limited. The difficulty in acquiring robustness makes these systems unreliable and often unfeasible in applications of high complexity. Furthermore, the unimodal approach (i.e. the observation of only one source of emotion) ignores the intrinsic relationships between these various affective inputs. Their relevance is studied in detail in [2].

Having such issues as motivation, this paper investigates the fusion of this diverse set of emotional components – or "sensors" – to determine whether the emotion shown by the interlocutor can by more precisely detected in a multimodal solution. Two classifiers are used for the evaluation of weights and relationships between the various inputs provided, which are then compared to unimodal solutions.

The paper presents some of the relevant work in this field in section 2, describes the tools used in section 3 and presents experiments in Sections 4 through 5.

2 Related Work

The formal modeling and understanding of human behavior is a widely discussed subject in psychology and neuroscience. Among the lines of thought, the Appraisal Theories, as discussed by Roseman [3] and Schorr [4], provide a model that explains the behavior differences of each unique individual at the same time that specifies aspects that are common to all. They describe the processes underlining emotion elicitation, which are the same for everybody, although their development is unique to the experience path of each individual. The quantification of these elicitations is also a vital topic, discussed by Sander [5].

Some of the first successful algorithms for emotion recognition on faces were described in [6]. In the decade that followed - with a substantial increase in computing power - approaches based on three-dimensional models of the face and optical flow in real time have become more common, improving over previous methods.

On the analysis of emotion in speech, one can cite recent advances in [7]. Some papers classify speech emotion through the valence in tone of the conversation (positive, neutral or negative), while others attempt to get more well-defined behaviors; a comparison may be found in [8].

On the use of the Internet as a database for emotional comprehension of words, some recent studies, like the one from [9], are starting to present encouraging results.

The multimodal analysis, in turn, still has a small and recent number of studies and publications. Besides [2], Campanella and Belin [10] perform an up-to-date discussion of cognitive studies, supporting the correlation between voice and facial expressions in the construct of emotion. In [11], data fusion of voice, facial and corporal expressions resulted in more than 10% of improvement over the unimodal approach, while [12] investigated the level at which the fusion should occur, using face and voice as inputs.

3 Tools

For facial expression analysis, the commercial software eMotion [13], developed at the University of Amsterdam, has been incorporated into the study. This choice was based on the good performance of the fitting algorithm for three-dimensional meshes and the ability to assess all the emotional states which are evaluated in this article.

For speech processing, Emo-Voice [14], developed by the Institute of Computer Sciences, University of Augsburg, was used. Emo-Voice is available as an open source package and allows for customized training of classifiers for detection of emotions in speech.

Finally, for the analysis of the emotional connotation of the speech, a dedicated tool was developed by the authors. Named “emoCrawler”, the tool will have its relevance evaluated in the experiments presented on this article. emoCrwaler collects the verbs, adjectives and nouns in the discourse and uses them in queries on social networks, evaluating the emotional reaction presented in the results, as described in the next section.

3.1 The EmoCrawler

As it was developed solely in the context of this paper, we discuss emoCrawler here. emoCrawler is a modular application currently in development, having as goal the evaluation of emotional contexts without the requirement of complex syntactic analysis. The program searches through the databases of online social networks, processing large volumes of results and analyzing the relative frequencies of emotional expressions. Thus, the program includes a dictionary inspired by the proposals of Goleman (apud [15]) and Laros [16], which maps different expressions to the emotions explored in this article.

The words expressed by the user which are considered significant are extracted from the discourse and are looked up in the social database. The messages related to that query are then indexed and checked against the existence of the emotional expressions contained in the dictionary. The algorithm works through a rule set, aiming at eliminating false positives. As an example, it considers the presence of negation elements in the treatment of queries.

For instance, let’s suppose the user inputs the phrase “I’m sorry that she died”. Since both keywords “sorry” and “died” do not define emotions, the software uses them in queries in the chosen social network. The resulting database is a set of messages that contain not only these words, but also the emotion words associated with them, which are then processed by the software. The application data flow is shown on Figure 1.

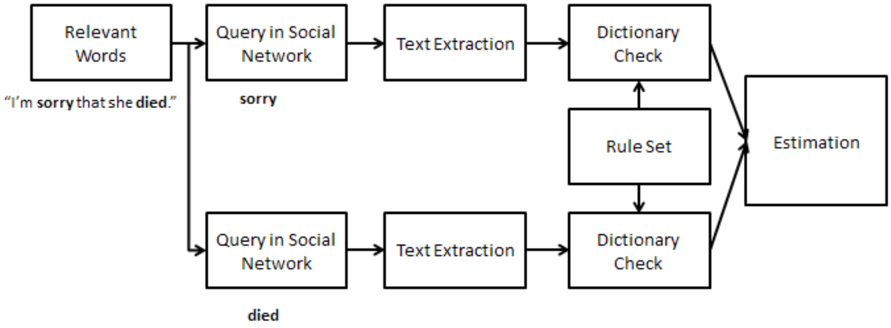


Fig. 1. Data flow example in emoCrawler

Given the modular context of emoCrawler, several social networks may be examined. For the present work, tough, only a sample from the microblog Twitter is considered [17]. The choice of Twitter as a data source is related to the conciseness and emotiveness of its posts. Users are frequently interested in expressing their opinion on a specific topic, but at the same time they have to do this in a summarized manner, without elaborate sentences – given the 140 character limitation.

The system must also consider the temporal factor in its analysis. The search for the emotional response connected to an element may present different results in different periods; the current popular opinion might diverge from the past. That being said, given the non-temporal context of the phrases used in the experiment, the mode of operation applied in this article queries only the current common sense regarding the elements of emotion.

4 Methodology

Having as objective the multimodal analysis, comprehending facial expression, voice and speech context, the *corpus eNTERFACE'05 Audio-Visual Emotion Database* [18] was selected. The database is a collection of videos comprised of scenes in which individuals are invited to express an emotional sentence in a way they feel is natural (Figure 2).

For the execution of experiments, three sets of samples that could be promptly classified by human observers were selected. The first set was reserved for the training of Emo-Voice, the second was used in the training of fusion classifiers and the third was selected for testing. For Emo-Voice, a SVM (Support Vector Machine) classifier was trained from samples of the corpus (20 for each emotion), allowing the algorithm to adapt to the characteristics of the recordings and language present in the database.

Even though there was a preliminary exclusion of unacceptable videos, many samples in non-ideal conditions were intentionally left in the experiment. Uneven lighting, audio noise and rapid movement of head and torso are some of the elements that were purposely left.

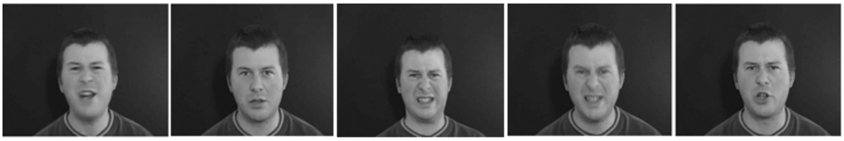


Fig. 2. Illustrative expressions presented in the corpus

Neural networks were used for the data fusion, as they are a common choice in situations where the input sources contain noise and in which the degree of relative reliability between them is unknown. Furthermore, these networks are also able to assess the importance of results provided by a single source, what has been used to evaluate the contribution of emoCrawler.

A subset of emotions classified by Ekman as “the big six” was used: happiness, sadness, fear, anger. This study left "surprise" out, because it is often understood as an expression without valence, not actually conveying an emotional state (i.e. surprise may be tied to any state). During the study, “disgust” was also removed from the set, for reasons discussed below.

Thus, the neural network has inputs originated from three different sources, one for each emotion from each source. The classifier also contains an output for every emotion, in order to perform the reverse path of nominal attribution, i.e., converting to "names of emotions". Even though some studies describe continuous transitions for some emotions, it would be impossible to map the output to a continuous scale in this particular case, as the presented emotions are not sortable in intensity or valence. Figure 3 illustrates the process.

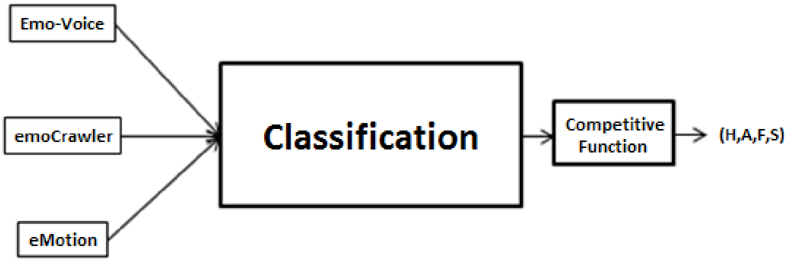


Fig. 3. Sensor Fusion

For the execution of experiments, two approaches were considered in regard to network topology and training algorithm: a feed-forward back propagation neural network (FFBPNN) and a probabilistic neural network (PNN). FFBPNN is a common solution for classification problems, while PNN, in turn, presents itself as a relevant alternative, given its training time orders of magnitude faster than the FFBPNN.

5 Experiments and Results

The first stage of evaluation consisted in the analysis of results coming directly from the sensors, i.e. the unimodal analysis. Table 1 shows the results of such tests for a heterogeneous set of samples, indicating the percentage of correct results by the tools.

Table 1. Rate of emotion detection (percentage of correct guesses) in unimodal analysis – heterogeneous sample set

Emotion	Face	Voice
Happiness	12.5%	12.5%
Anger	88.9%	11.1%
Disgust	0%	0%
Fear	50,0%	50,0%
Sadness	50,0%	100,0%

As it can be seen from Table 1, both sensors, emoVoice and eMotion, had a very bad performance in detecting “disgust”, possibly due to the amateur profile of the actors. Therefore, this emotion has been removed from the classifier training and test sets, to avoid contamination issues in the classification of the other four emotions.

In a second step, data from the remaining four emotions was separated into training and test sets for the neural networks, with equal number of samples for each emotion. FFBPNN learning consisted in the application of the Robust Backpropagation algorithm to the training set. The choice of the amount of hidden layer nodes was performed by studying the rates of convergence of the network.

The probabilistic network adjustment, in turn, relies heavily on the smoothing parameter, a positive scalar related to the distance between training vectors. The methodology for choosing such parameter was to start it with a high value (generalist) and to perform successive reduction steps to ensure full compliance of training data in simulation, resulting in a spreading factor of 0.17 as the best match.

Table 2 shows, after the supervised learning, the results for the test set, comparing the unimodal assessment of emotions in face and voice with the data coming from multimodal fusion. Since emoCrawler is inherently associated with the classifier, it was not evaluated in a unimodal approach.

Table 2. Comparison of individual measurements against multimodal fusion: rate of detection for each method

Emotion	Voice	Face	FFBP Fusion (face/voice/semantics)	PNN Fusion (face/voice/semantics)
Happiness	20%	0%	60%	60%
Anger	100%	0%	100%	100%
Fear	40%	20%	80%	60%
Sadness	100%	60%	60%	60%
Average Rate	**65%**	**20%**	**75%**	**70%**

To assess the influence of emoCrawler, a bimodal FFPBNN was trained, using only face and voice information. Table 3 presents the results when submitted to the test set.

Table 3. Evaluation of emoCrawler's efficiency over the FFBPNN test group: rate of detection in each case

Emotion	emoCrawler disabled	emoCrawler enabled
Happiness	20%	60%
Anger	60%	100%
Fear	20%	80%
Sadness	100%	60%
Average Rate	**50%**	**75%**

It's possible to observe the superior multimodal behavior when semantic comprehension elements are present, particularly in emotions in which the face and voice systems had poor performance. Despite the coherent results, there's a decrease in accuracy in the specific case of sadness, likely a consequence of intense noise in emoCrawler's training dataset, which caused confusion in the learning process.

6 Final Remarks and Future Work

In realistic scenarios, emotion detection based on facial analysis provides somewhat poor results due to uneven lightning and body movement, but mostly because speech causes facial deformations that may be interpreted as different emotions. Voice-only emotion detection is strongly influenced by valence. A multimodal approach, as the one discussed in this work, can outperform a unimodal system, as shown by experimental results.

The addition of semantic information through emoCrawler, a tool built by the authors to search for emotional content of words on Internet social networks, was demonstrated to improve emotion detection over bi-modal (face+speech) systems.

As future work, the evaluation of different classifiers (particularly Bayesian Networks) and the comparative analysis in regard to the results of this paper is planned. It's also an upcoming goal to use multimodal fusion in situations where even longer (several minutes) human-machine interactions are present, perhaps considering windowing for a closer relationship with reality.

Acknowledgments. The authors would like to thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), the Department of Mechatronics Engineering at EPUSP and FAPESP (by support through process n. 2008/03995-5) for the collaboration and financial support in this research.

References

1. Ekman, P., Friesen, W.: Facial Action Coding System. Consulting Psychologist Press (1977)
2. Scherer, K., Ellgring, H.: Multimodal Expression of Emotion: Affect Programs or Componential Appraisal Patterns? American Psychological Association (2007)
3. Roseman, I.J., Smith, C.: Appraisal Theory – Overview, Assumptions, Varieties, Controversies. In: Scherer, K., Schorr, A., Johnstone, T. (eds.) Appraisal Processes in Emotion – Theory, Methods, Research. University Press, USA (2001)
4. Schorr, A.: Appraisal – The Evolution of an Idea. In: Scherer, K., Schorr, Johnstone, T. (eds.) Appraisal Processes in Emotion – Theory, Methods, Research. Oxford University Press, UK (2001)
5. Sander, D., Grandjean, D., Scherer, K.R.: A systems approach to appraisal mechanisms in emotion. Neural Networks 18, 317–352 (2005)
6. Bartlett, M.S., Hager, J.C., Ekman, P., Sejnowski, T.J.: Measuring facial expressions by computer image analysis. Department of Cognitive Science, University of California, San Diego, USA (1999)
7. Rachuri, K.K., Musolesi, M., Mascolo, C., Rentfrow, P., Longworth, C., Aucinas, A.: EmotionSense: a mobile phone based adaptive platform for experimental social psychology research. In: UbiComp 2010, Copenhagen, Denmark, September 26-September 29 (2010)
8. Vogt, T., André, E.: Comparing Feature Sets for Acted and Spontaneous Speech in View of Automatic Emotion Recognition. In: IEEE International Conference on Multimedia & Expo, ICME 2005 (2005)
9. Ptaszynski, M., Dybala, P., Shi, W., Rzepka, R., Araki, K.: Towards Context Aware Emotional Intelligence in Machines: Computing Contextual Appropriateness of Affective States. In: IJCAI 2009 Proceedings of the 21st International Jont Conference on Artifical Intelligence (2009)
10. Campanella, S., Belin, P.: Integrating face and voice in person perception. Trends in Cognitive Sciences 11, 535–543 (2007)
11. Castellano, G., Kessous, L., Caridakis, G.: Multimodal emotion recognition from expressive faces, body gestures and speech. In: de Rosis, F., Cowie, R. (eds.) Proc. of the Doctoral Consortium of 2nd International Conference on Affective Computing and Intelligent Interaction, Lisbon (September 2007)
12. Chetty, G., Wagner, M.: A Multilevel Fusion Approach for Audiovisual Emotion Recognition. In: International Conference on Auditory-Visual Speech Processing (2008)
13. eMotion - Visual Recognition, `http://www.visual-recognition.nl` (accessed March 2011)
14. Vogt, T., André, E., Bee, N.: EmoVoice - A framework for online recognition of emotions from voice. In: Proceedings of Workshop on Perception and Interactive Technologies for Speech-Based Systems (2008)
15. Martinez-Miranda, J., Aldea, A.: Emotions in human and artificial intelligence. Computers in Human Behavior 21, 323–341 (2005)
16. Laros, F.J.M., Steenkamp, J.E.M.: Emotions in consumer behavior: a hierarchical approach. Journal of Business Research 58, 1437–1445 (2005)
17. Twitter – The best way to discover what's new in your world, `http://www.twitter.com` (accessed March 2011)
18. Martin, O., Kotsia, I., Macq, B., Pitas, I.: The eNTERFACE'05 Audio-Visual Emotion Database. Université Catholique de Louvain, Aristotle University of Thessaloniki (2005)

Improving the MLP Learning by Using a Method to Calculate the Initial Weights of the Network Based on the Quality of Similarity Measure

Yaima Filiberto Cabrera[1], Rafael Bello Pérez[2],
Yailé Caballero Mota[1], and Gonzalo Ramos Jimenez[3]

[1] Department of Computer Sciences, University of Camaguey
{yaima.filiberto,yaile.caballero}@reduc.edu.cu
[2] Department of Computer Sciences, Central University of Las Villas
rbellop@uclv.edu.cu
[3] Department of Languages and Computer Science, University of Malaga
ramos@lcc.uma.es

Abstract. This work presents a technique that integrates the backpropagation learning method with a method to calculate the initial weights in order to train the Multilayer Perceptron Model. The method to calculate the initial weights of the MLP is based on the quality of similarity measure proposed on the framework of the extended Rough Set Theory. Experimental results show that the proposed initialization method performs better than other methods used to calculate the weight of the features, so it is an interesting alternative to the conventional random initialization.

Keywords: Multilayer perceptron, weight initialization, quality of similarity measure.

1 Introduction

Artificial Neural Networks (ANN) constructs a functional relationship between input patterns and output patterns through the learning process; they memorize the relationship in the form of weights, and recall it in applications. Neural networks must have a good generalization capability to produce correct outputs for untrained patterns. The possibility to learn and the generalization capabilities allow applying the ANN to various problems that are hardly solved by conventional techniques. Each neural network is determined by its structure and the learning algorithm. Learning algorithms adjust network weights in a way that the behavior of the network improves after each iteration. In supervised problems, learning algorithms are based on the output error; this error is the difference between the neural network output and the desired output, and this is a function of the weights; algorithms minimize the output error by adjusting neural networks' weights [1]. This provides a powerful base learner, with advantages such as nonlinear mapping and noise tolerance. These are increasingly used in the Data Mining and Machine Learning fields due to their good behavior in terms of predictive knowledge.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 351–362, 2011.

The crucial part in ANN-based modeling is ANN training. Many learning algorithms are used for the neural networks to learn the relationship between input patterns and output patterns; among them, the backpropagation (BP) algorithm is most widely used to train the Multilayer Perceptron (MLP), which is one of the most popular models in artificial neural networks, where neurons are grouped in layers and only forward connections exists. The basic idea of the BP algorithm is that weights are updated in the direction to reduce the error between the desired output and the actual output. The essential character of the BP algorithm is gradient descent. Because the gradient descent algorithm is strictly dependent on the shape of the error surface, the error surface may have some local minimum and multimodal. Therefore, some problems of this method are: falling into some local minimum, premature convergence, and slow convergence speed [2, 3].

The success of ANN usage in fields such as pattern recognition, signal processing, etc. is incontestable, but a key problem concerning neural networks usage in practice remains as a challenge. This problem is related to correctly building neural networks, specially tailored to a specific problem. That is, the difficult work regarding the correct specification (tuning) of parameters for a given problem, including the initial weights, according to [4]. It is the case of MLP. The MLP can approximate a wide range of functions [5]; but, careful control must be exercised over the complexity of the MLP to avoid overfitting and underfitting. To obtain successful use, the network topology and the initial weights play a very important role. In general, the training of the MLP neural networks is accomplished through successive attempts with different network topologies and sets of weights until reaching satisfactory results for the problem.

A key point for a neural net is: How do we compute the neural net weights? This includes an initialization algorithm and a training algorithm. The weights can be randomly initialized, but it is important to take into account that the ANN performance will heavily depend on the value [6]; this is an old but also a nowadays' problem, such as old and new studies show [2, 3, 7-14], etc.

Studies have shown that one of the main reasons for the slow convergence and the suboptimal generalization results of MLP based on gradient descent training is the lack of a proper initialization of the weights to be adjusted. Nevertheless, sophisticated learning procedures are not yet able to compensate for bad initial values of weights, while a good initialization would lead to a convergence and/or quick generalization capability even with a simple gradient-based error minimization technique [13].

There are ongoing efforts to integrate artificial neural networks with the Fuzzy Set Theory, Genetic algorithms, Rough Set Theory and other methodologies in the Soft Computing paradigm. This is the case of this work, in which the problem of weight initialization in MLP is considered. A weight initialization method based on the quality of similarity measure proposed on the framework of the Rough Set Theory (described in section 3) is investigated, which is presented in section 4. A comparison with the conventional random initialization and other methods shows that significant improvement in convergence can be achieved with the method proposed in this paper; see section 5.

2 About Multilayer Perceptron and Its Learning Process

Kindly the MLP, a feedforward ANN with sigmoidal activation trained by BP [15], is one of the most used connectionist models in the literature. The training algorithms minimize an error function by tuning the modifiable parameters of a fixed architecture, which needs to be set a priori. Thus, the correct design of the MLP topology is a complex and crucial task, commonly addressed by trial-and-error procedures. In addition, the gradient-based procedures used for the MLP training are not free from getting trapped into local minima when the error surface is rugged, being also sensitive to parameter settings and to the network initial weights [16].

ANNs such as MLP are powerful models for solving nonlinear mapping problems. The training of ANNs is a nonlinear optimization problem in which the goal is to find a set of weight that minimizes the cost function. The cost function is usually a function of the network mapping errors. In addition, the cost function is usually characterized by a large number of local minima in the vicinity of the global minimum.

The training set (TS) for the network consists of data samples. Each data sample consists of an input vector and desired output vector *(X_p, Yp)*, where p denotes the index of the sample; in the case of the function approximation problem Y_p is usually a single real value, while in classification problems Y_p is a single discrete value.

Suppose there is a set of training patterns $TS=\{(X_1,Y_1), (X_2,Y_2), \ldots, (X_l,Y_l)\}$ where l is the number of training patterns, each training pattern X_i in set TS is an m-dimensional feature vector. The training of the ANN means minimizing an error function with respect to the weights W and is minimized when the network output matches the desired outputs. The set of weights W represents the knowledge of the application domain, so the key task while an ANN is training is to find the right set W; which is especially relevant in the case of MLP network.

BP training suffers the fact of being very sensitive to initial conditions. In general terms, the choice of the initial weight vector w_0 may speed convergence of the learning process towards a global or a local minimum if it happens to be located within the attraction basin of that minimum. Conversely, if w_0 starts the search in a relatively flat region of the error surface it will slow down the adaptation of the connection weights. Sensitivity of BP to initial weights, as well as to other learning parameters, was studied experimentally by many researches, such as [7-10]. They have shown that a proper weight initialization can facilitate in a very significant way the training process.

Conventionally, the rule employed for the weight initialization is to use small random values; usually the initial weights of the MLP take random values distributed uniformly on small intervals. The motivation for this is that large absolute values of weights can cause networks nodes to be highly active or inactive for all training samples, and thus insensitive to the training process. However, too small values can also slow down the training. This conventional initialization can be adequate in many simple problems. However, in more difficult problems it has been found that more deterministic initializations can improve the convergence of training drastically compared with random initialization. For instance, in [2] authors stated that when all the weights are set to random values, the learning curves have a typical feature in multiclass classification problems; after decreasing rapidly in the beginning, the squared error sum maintains a near constant value for a long time, weights get into a flat area where gradients are very small. According to these authors, these phenomena make the MLP net works show a worse performance in the case of multiclass classification problem than in two class classification problems.

Weight initialization tries to provide initial weight values close to the global minimum of the error function, in order to avoid local minima. There are several strategies for initializing the MLP weights. In [11] several ANN weight-initialization methods were introduced and compared mainly by means of classification problems. It was shown how the choice of an initialization method influences the convergence of the optimization. One way to ensure that we get a good result is to use a multi-start algorithm, which consists of using several random initializations and choosing one of them, or a combination between them; that is, several neural nets are trained for each frame, and the neural net that yields the smaller training error is chosen [6]; also authors in [14] consider some random starting points into the weight space (that is, they introduce multiple starting points), and try, by means of some extra control, to hit at least one location that feels promising.

3 Building Similarity Relations Based on the Quality of Similarity Measure

In [17], a method is proposed for building relations of similarity in the context of the extended Rough Set Theory, in which the equivalence relation classic RST [18] is replaced by a weaker binary relation [19-21]. This method works on the principle that "similar problems have similar solutions". The similarity between two objects *(x, y)* from the description of the problems (according to n condition features) and the similarity regarding the solution of the problems (feature of decision) is set according to the relations R_1 and R_2 respectively, as defined by the expressions (1) and (2):

$$xR_1 y \textit{ if and only if } F_1(x,y) \geq e_1 \tag{1}$$

$$xR_2 y \textit{ if and only if } F_2(x,y) \geq e_2 \tag{2}$$

The function F2 is a function for comparing the values of the feature decision. Note that two objects in classification problems are similar objects according to F_2 if they belong to the same class; e_1 and e_2 are thresholds, and the function F_1 is defined by the expression (3):

$$F_1(x,y) = \sum_{i=1}^{n} w_i * \partial_i(x_i, y_i) \tag{3}$$

where w_i is the weight of feature i and $\delta_i()$ is the comparison function for the feature i. The purpose is to find relations R_1 and R_2 such that similar objects to any object x of the universe according to R_1 and R_2 are practically the same, i.e. the set N_1 and N_2, defined by the expressions (4) and (5), should be similar (optimal case is $N_1(x) = N_2(x)$, for any object x).

$$N_1(x) = \{y : x R_1 y\} \tag{4}$$

$$N_2(x) = \{y : x R_2 y\} \tag{5}$$

The degree of similarity between both sets for an object x is expressed by the following measure:

$$\varphi(x) = \frac{|N_1(x) \cap N_2(x)|}{0.5 * |N_1(x)| + 0.5 * |N_2(x)|} \quad 0 \le \varphi(x) \le 1 \tag{6}$$

From which is defined, by the expression (7), the quality of the similarity measure of a decision system *(DS)* with a universe of M objects:

$$\theta(DS) = \left\{ \frac{\sum_{i=1}^{M} \varphi(x)}{M} \right\} \tag{7}$$

The objective is to maximize the value of $\theta(DS)$. The value of this measure depends on the function F_1. Using the weighted sum defined by (3), and given the comparison functions for each feature, the problem reduces to finding the set of weights $W = \{w_1, w_2, ..., w_n\}$, to construct F_1. To find the set of weights W associated with the condition features, a method is proposed in [17], this method uses the metaheuristic Particle Swarm Optimization (PSO) [22], the particles are real vectors with dimension n, and the evaluation function of the particles is defined by (7). At the end of the search process, the particle with maximum value of the measure quality of similarity is used as weights for the condition features.

4 A Method to Calculate the Initial Weights in the MLP Network

Our proposal consists of using the set of weights W as the initial weights between the input layer and the hidden layer in the MLP network.

In general, MLPs can have several hidden layers. However, we consider the initialization of MLPs with only one hidden layer. Assuming a three-layer neural network with n inputs (features), q outputs (categories), and one hidden layer with a variable number of nodes $(n+q)/2$; see Fig. 1.

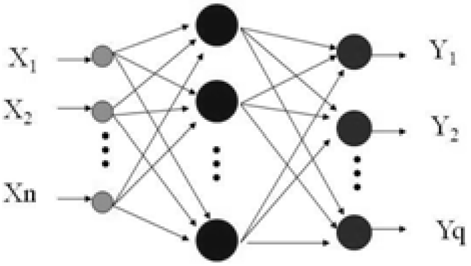


Fig. 1. The topology of the MLP in this research

The initial values for the MLP are the following:

- W_{ji} (weight of the connection between the *jth* hidden node and the *ith* input node) are assigned in the following way: $W_{ji}=W_i$ for all *j*, where W_i is the calculated weight for the feature *i*. Note, all links from the *ith* input node have the same value (the weight calculated for the *ith* feature using the method described in section 3).
- V_{kj} (weight the connection between the *jth* hidden node and the *kth* output node) are random values in the interval [0, 1].

5 Experimental Results

The experiments include problems of classification and problems of approximation of functions. There were used 24 databases from the UCI repository [23], 12 where the domain of the decision attributes is nominal (classification), and 12 where the domain of the decision attributes is numerical (approximation of functions), Tables 1 and 2.

Table 1. Databases with discrete decision attribute

BD	Instances	Attributes
Tae	151	5
Diabetes	768	8
Iris	150	4
Hepatitis	155	19
postoperative-patient-data	90	8
zoo	101	17
bridges-version1	107	12
Biomed	194	8
Schizo	104	14
soybean-small	47	35
Cars	392	7
heart-statlog	270	14

Table 2. Databases with decision attribute with continuous domain

DB	Instances	Attributes
baskball	96	4
bodyfat	252	14
detroit	13	13
diabetes_numeric	43	2
elusage	55	2
fishcatch	158	7
pollution	60	15
pwLinear	200	10
pyrim	7	27
sleep	51	7
vineyard	52	3
schlvote	37	5

For the experimental study the training and control sets were obtained, taking for the first one 75% of the cases and for the second 25%, in a completely random way. Following this principle of random selection repeated the process k times and they were obtained for each datasets k sets of training and k control sets, with the purpose of applying cross validation [25] for a better validation of the results. The value of k that was used is 10, as recommended in [25].

We performed an experimental study of the accuracy of the results achieved by MLP model used as function approximator.

In the approximation problem, the purpose is to find the relation $Y=\varphi(xq)+\varepsilon$; here, $\varphi(xq)$ is known as the regression function of y on xq, and ε is a random noise process; regression modeling is concerned with estimation of the regression function given a set of n noise contaminated samples of this function. In parametric regression modeling, it is assumed that $\varphi(x)$ can be written in some parametric form $f(x;\theta)$, for some values of the parameters θ, such as a polynomial with adjustable coefficients (parameters). The MLP implements a parametric model and it fits into the nonlinear regression framework [26].

Four different methods were used to initialize the weights in the MLP: (i) random generation (MLP-Ram), (ii) calculation of the weights by the conjugate gradient method (KNN-$_{\mathrm{VSM}}$) [27], (iii) using the same weight value for all traits (Stand = 1/numAtt), and (iv) the weights calculated by the method proposed in this paper (PSO + RST). The comparative study of the results was performed using the correlation coefficient (Coef-Corr), the coefficient R2 (Coef-R2) and two measures: (i) Mean Absolute Percentage Error, MAPE and (ii) The average of the differences between the desired and produced value by the method (PMD). These measures are defined by expressions (8) - (9):

$$MAPE = \frac{\sum_{i=1}^{M} \left| \frac{ai - yi}{ai} \right|}{M} * 100\% \tag{8}$$

$$PMD = \frac{\sum_{i=1}^{M} \left| ai - yi \right|}{M} \tag{9}$$

Where:
ai is the output value.
yi is the value produced by the method .
M is the number of objects.

The results achieved by the MLP for the case of approximation of functions when initializing the weights using the above variants are shown in Tables 3 and 4 where you can observe the better performance of the proposed method PSO + RST.

Table 3. Acuracy obtained by each method according to PMD and MAPE

	PMD				**MAPE**			
			MLP-	**PSO+**			**MLP-**	**PSO+**
DB	**Stand.**	**KNN$_{VSM}$**	**Ram**	**RST**	**Stand.**	**KNN$_{VSM}$**	**Ram**	**RST**
baskball	0.083	0.083	0.082	0.079	22.218	22.218	22.090	20.953
bodyfat	0.540	0.540	2.128	0.519	5.548	5.548	12.572	4.722
detroit	32.778	32.778	33.058	25.646	9.870	9.870	10.094	7.857
diabetes_numeric	0.528	0.528	0.528	0.531	11.894	11.894	11.914	11.888
elusage	10.670	10.766	10.758	10.367	27.397	28.238	26.457	26.651
fishcatch	47.927	47.927	38.602	33.989	30.481	30.481	36.235	33.791
pollution	42.407	42.408	47.393	43.360	4.516	4.516	5.144	4.599
pwLinear	2.072	2.072	1.681	1.665	245.232	245.274	121.661	215.564
pyrim	0.095	0.095	0.083	0.077	19.810	19.812	18.612	16.130
sleep	3.174	3.174	3.279	2.889	40.476	40.476	39.344	36.946
vineyard	2.410	2.410	2.093	2.103	22.521	22.520	15.750	15.918
schlvote	991920	991920	332232	332232	194	194	158	171

Table 4. Accuracy obtained by each method according to the correlation coefficient and the R2 coefficient

	Corr- Coef				**R2- Coef**			
DB	**Stand.**	**KNN$_{VSM}$**	**MLP-Ram**	**PSO+RST**	**Stand.**	**KNN$_{VSM}$**	**MLP-Ram**	**PSO+RST**
baskball	0.467	0.467	0.460	0.516	0.212	0.212	0.212	0.266
bodyfat	0.996	0.996	0.995	0.996	0.974	0.974	0.971	0.975
detroit	0.854	0.854	0.830	0.888	0.729	0.729	0.688	0.789
diabetes_numeric	0.484	0.484	0.478	0.506	0.235	0.235	0.229	0.256
elusage	0.814	0.813	0.834	0.843	0.663	0.662	0.696	0.710
fishcatch	0.987	0.987	0.989	0.990	0.957	0.957	0.978	0.980
pollution	0.637	0.637	0.549	0.707	0.406	0.406	0.301	0.500
pwLinear	0.840	0.840	0.893	0.900	0.652	0.652	0.774	0.794
pyrim	0.452	0.452	0.589	0.694	0.204	0.204	0.347	0.481
sleep	0.635	0.635	0.616	0.684	0.403	0.403	0.380	0.468
vineyard	0.671	0.671	0.796	0.801	0.451	0.451	0.634	0.642
schlvote	0.199	0.199	0.151	0.428	0.040	0.040	0.023	0.183

In the case of using the MLP method for classification problems, also performed an experimental study using various methods to initialize the weights, they are: (i) when the weights are initialized with random values (MLP-Ram), (ii) when the weights are calculated using the conjugate gradient method (KNN-VSM) [27], (iii) when the weights are calculated by using the method Relieff (RELIEF) [28], (iv) giving the same weight value to all traits (Stand = 1/numAtt), (v) and weights calculated by the proposed method (PSO + RST). The results achieved by the MLP for the case of classification where the weights are initialized using the mentioned variants are shown in Table 5, where you can see the best performance of the method PSO + RST.

Table 5. Results of the classification's general accuracy

DB	**PSO+RST**	**Stand.**	**KNN$_{VSM}$**	**RELIEF**	**MLP-Ram**
Tae	58.94	49.01 ⊖	55.63 ⊖	54.97 ⊖	54.30 ⊖
Diabetes	76.17	76.69 ⊕	74.22 ⊖	74.74 ⊖	75.39 ⊖
Iris	98.00	95.33 ⊖	96.67 ⊖	98.00 ⊖	97.33 ⊖
Hepatitis	84.52	78.06 ⊖	81.29 ⊖	79.35 ⊖	80.00 ⊖
postoperative-patient-data	57.78	54.44 ⊖	53.33 ⊖	55.56 ⊖	55.56 ⊖
zoo	96.04	73.27 ⊖	40.59 ⊖	75.25 ⊖	94.29 ⊖
bridges-version1	71.43	41.90 ⊖	41.90 ⊖	60.00 ⊖	69.52 ⊖
Biomed	92.78	83.51 ⊖	82.99 ⊖	83.51 ⊖	86.08 ⊖
Schizo	68.27	63.46 ⊖	62.50 ⊖	63.46 ⊖	65.38 ⊖
soybean-small	100	78.72 ⊖	76.60 ⊖	74.47 ⊖	100 ⊙
Cars	80.10	71.17 ⊖	71.17 ⊖	71.17 ⊖	78.06 ⊖
heart-statlog	81.85	80.37 ⊖	80.37 ⊖	80.37 ⊖	78.15 ⊖

In order to compare the results, we will use a multiple comparison test to find the best algorithm. In Table 6 can be observed that the best ranking is obtained by our proposal. Thus, ⊕ indicates that the accuracy is significantly better than the accuracy achieved when PSO+RST is not used, ⊖ signifies that the accuracy is significantly worse and ⦿ signifies that there is no significant differences.

An Iman–Davenport test is carried out (employing F-distribution with 4 and 44 degrees of freedom) in order to find statistical differences among the algorithms, obtaining a p-value near to zero. In this manner, Table 7 shows the results of the Holm procedure for comparing our proposal to the remaining ones. The algorithms are ordered with respect to the obtained z-value.

Thus, by using the normal distribution, we can obtain the corresponding p-value associated with each comparison and this can be compared with the associated α/i in the same row of the table to show whether the associated hypothesis of equal behavior is rejected in favor of the best ranking algorithm, as we can observe, the test rejects all cases. It can be noticed that this approach is statistically superior to all compared methods.

Table 6. Average ranks obtained by each method in the Friedman test

Algorithm	Ranking
PSO+RST	1.1667
Est	3.7917
KNN_{VSM}	4.0417
RELIEF	3.3333
MLP-Ram	2.6667

Table 7. Holm's table for α = 0.05, PSO+RST is the control method

i	Algorithm	z=(R0-Ri)/SE	p	Holm /Hochberg /Hommel	Hypothesis
4	KNN_{VSM}	4.453931	0.000008	0.0125	Reject
3	Stand.	4.066633	0.000048	0.016667	Reject
2	RELIEF	3.356586	0.000789	0.025	Reject
1	MLP-Ram	2.32379	0.020137	0.05	Reject

6 Conclusions

In this paper we revisit the problem of weight initialization for feedforward neural networks trained with gradient descent based procedures. To overcome the problem due to the random initialization, a new initialization method is proposed for training the MLP network. The basic idea of the proposed method is to set the initial weights from the input layer to the hidden layer using the weights of conditional features which are calculated to build the similarity relation that maximize the measure quality of similarity proposed in the framework of the Rough Set Theory. The experimental study for problems of function approximation and classification shows a superior performance of the MLP when the weights are initialized using the method proposed in this work, compared to other previously reported methods to calculate the weight of features.

References

1. Hocenski, Z., Antunovic, M., Filko, D.: Accelerated Gradient Learning Algorithm for Neural Network Weights Update. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part I. LNCS (LNAI), vol. 5177, pp. 49–56. Springer, Heidelberg (2008)
2. Kim, M., Choi, C.: A New Weight Initialization Method for the MLP with the BP in Multiclass Classification Problems. Neural Processing Letters 6, 11–23 (1997)
3. Fu, X., Zhang, S., Pang, Z.: A Resource Limited Immune Approach for Evolving Architecture and Weights of Multilayer Neural Network. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 328–337. Springer, Heidelberg (2010)
4. Almeida, L.M., Ludermir, T.B.: An Evolutionary Approach for Tuning Artificial Neural Network Parameters. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 156–163. Springer, Heidelberg (2008)
5. Hornik, K., et al.: Multilayer feedforward neural network are universal approximators. Neural Networks 2, 359–366 (1989)
6. Faúndez-Zanuy, M.: Nonlinear Speech Processing: Overview and Possibilities in Speech Coding. In: Chollet, G., Esposito, A., Faúndez-Zanuy, M., Marinaro, M. (eds.) Nonlinear Speech Modeling and Applications. LNCS (LNAI), vol. 3445, pp. 15–42. Springer, Heidelberg (2005)
7. Kim, Y.K., Ra, J.B.: Weight value initialization for improving training speed in the backpropagation networks. In: Kim, Y. (ed.) IEEE International Joint Conference on NN, pp. 2396–2401 (1991)
8. Drago, G., Ridella, S.: Statistically controlled activation weight initialization. IEEE Transaction on Neural Networks 3(4), 627–631 (1992)
9. Wessels, L., Barnard, E.: Avoiding false local minima by proper initialization. IEEE Transaction on Neural Networks 3(6), 899–905 (1992)
10. Denoeux, T., Lengelle, R.: Initializing back propagation networks with prototypes. Neural Networks 6, 351–363 (1993)
11. Thimm, G., Fiesler, E.: High-order and multilayer perceptron initialization. IEEE Trans. on Neural Networks 2, 349–359 (1997)
12. Almeida, L.M., Ludermir, T.B.: An Evolutionary Approach for Tuning Artificial Neural Network Parameters. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 156–163. Springer, Heidelberg (2008)
13. Adam, S., Karras, D.A., Vrahatis, M.N.: Revisiting the Problem of Weight Initialization for Multi-Layer Perceptrons Trained with Back Propagation. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008. LNCS, vol. 5507, pp. 308–315. Springer, Heidelberg (2009)
14. Nieminen, P., Kärkkäinen, T.: Ideas about a Regularized MLP Classifier by Means of Weight Decay Stepping. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 32–41. Springer, Heidelberg (2009)
15. Rumelhart, D.E., Hilton, G.E., Williams, R.J.: Learning Representations by Backpropagation Errors. Nature 323, 533–536 (1986)

16. Rocha, M., Cortez, P.C., Neves, J.: Simultaneous Evolution of Neural Network Topologies and Weights for Classification and Regression. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 59–66. Springer, Heidelberg (2005)
17. Filiberto, Y., Bello, R., Caballero, Y., Larrua, R.: Using PSO and RST to Predict the Resistant Capacity of Connections in Composite Structures. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) NICSO 2010. SCI, vol. 284, pp. 359–370. Springer, Heidelberg (2010)
18. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
19. Skowron, A.: Logic, algebra and computer science, Helena Rasiowa and Cecylia Rauszer in Memoriam. In: Bulletin of the Section of Logic, pp. 1–215 (1996)
20. Slowinski, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Data and Knowledge Engineering 12(2), 331–336 (2000)
21. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177, 28–40 (2007)
22. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. IEEE Service Center, Piscataway (1995)
23. Asuncion, A., Newman, D.: UCI machine learning repository. A study of the behaviour of several methods for balancing machine learning training data. SIGKDD Explorations 6(1), 20–29 (2007)
24. Mitchell, T.: Machine Learning. In: Science/Engineering/Math., McGraw Hill, Portland (1997)
25. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research (7), 1–30 (2006)
26. Myles, A., et al.: Estimating MLP generalisation ability without a test set using fast, approximate Leave-one-out cross-validation. Neural Computing and Applications 5, 134–151 (1997)
27. Wettschereckd, D.: A description of the mutual information approach and the variable similarity metric. Technical report, Artificial Intelligence Research Division. German National Research Center for Computer Science, Sankt Augustin (1995)
28. Kononenko, I.: Estimating attributes: Analysis and extensions of RELIEF. In: Proc. European Conf. on Machine Learning (1994)

Modular Neural Networks with Type-2 Fuzzy Integration for Pattern Recognition of Iris Biometric Measure

Fernando Gaxiola, Patricia Melin, Fevrier Valdez, and Oscar Castillo

Tijuana Institute of Technology, Tijuana México
fergaor_29@hotmail.com,
{pmelin,fevrier,ocastillo}@tectijuana.mx

Abstract. This paper presents a new modular neural network architecture that is used to build a system for pattern recognition based on the iris biometric measurement of persons. In this system, the properties of the person iris database are enhanced with image processing methods, and the coordinates of the center and radius of the iris are obtained to make a cut of the area of interest by removing the noise around the iris. The inputs to the modular neural network are the processed iris images and the output is the number of the identified person. The integration of the modules was done with a type-2 fuzzy integrator at the level of the sub modules, and with a gating network at the level of the modules.

Keywords: Type-2 Fuzzy Logic, Modular Neural Network, Iris Biometric, Pattern Recognition.

1 Introduction

This paper is focused on the area of modular neural networks 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 a modular neural network divided into three modules, each module's input is a part of the database of human iris, and some methods or techniques are used for pre-processing the images such as normalization, resizing, cut, edge detection, among several others. Each module was divided into two sub modules with different neural networks.

The rest of the paper is organized as follows: Section 2 contains a brief explanation from previous research with human iris for recognition and basic concepts relevant to the area, section 3 defines the method proposed for this research and the description of problem addressed in this paper, section 4 presents the results achieved in this work and Section 5 draws conclusions and future work.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 363–373, 2011.

2 Background and Basic Concepts

2.1 Modular 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 modular neural networks are composed of simpler networks that behave as functional blocks and these are the neural modules.

A modular neural network works similarly to a classical neural network, as it is composed of sigmoidal activation neurons or linear, discrete and trains with common learning algorithms (gradient descent with adaptive learning algorithm, backpropagation, gradient descent scaling , etc.). What distinguishes it from other neural models is that it develops based functional modules and each module runs a neural network with the same characteristics or different (input layer, hidden layers, output layer, depending activation, learning algorithm, number of neurons per layer, etc.). In this model the modules work independently and in the end a form commonly called integrator performs the function of deciding between the different modules to determine which of them has the best solution (including gating networks, fuzzy integrator, etc.). [6]. Figure 1 shows a modular neural network scheme:

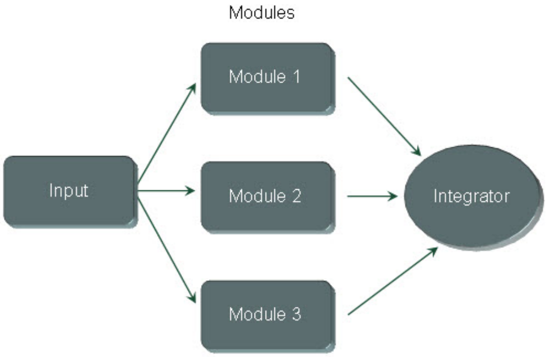


Fig. 1. Schematic of an modular artificial neural network

2.2 Fuzzy Logic

Fuzzy logic is an area of soft computing that enables a computer system to reason with uncertainty [7]. 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 [8]. 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 [9]. 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 [10][11][12].

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 [13][14][15][16][17][18].

2.4 Historical Development

The first use of the iris was presented in Paris, where criminals were classified according to the color of their eyes following a proposal by the French ophthalmologist Bertillon (1880) [19]. Research in visual identification technology began in 1935. During that year an article appeared in the 'New York State Journal of Medicine', which suggested that "the pattern of arteries and veins of the retina could be used for unique identification of an individual" [20].

After researching and documenting the potential use of the iris as a tool to identify people, ophthalmologists Flom and Safir patented their idea in 1987 [21]; and later, in 1989, they patented algorithms developed with the mathematician Daugman. Thereafter, other authors developed similar approaches [20].

In 2001, Daugman also presented a new algorithm for the recognition of people using the biometric measurement of Iris [22].

In 2005, Roy proposes an iris recognition system for the identification of persons using support vector machine [23].

In 2006, Cho and Kim presented a new method to determine the winner in LVQ neural network [24].

In 2009, Sarhan used the discrete cosine transform for feature extraction and artificial neural network for recognition [25]; Abiyev and Altunkaya presented the iris recognition system using neural network [26].

The literature has well documented the uniqueness of visual identification. The iris is so unique that there are no two irises alike, even twins, in all humanity. The probability of two irises producing the same code is 1 in 10^{78}, becoming known that the earth's population is estimated at approximately 10^{10} million [27], it is almost impossible to occur.

3 Proposed Method and Problem Description

This work focuses primarily on the identification of individuals. This problem is well known by the scientific community, as innumerable investigations have been developed 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).

The specific problem considered in this work is: "Obtain a good percentage of person identification based on the biometric measurement of the human iris, using modular neural networks”.

We used a database of human Iris from the Institute of Automation of the Chinese Academy of Sciences (CASIA) (see Figure 2). It consists of 14 images (7 right eye - 7 left eye) per person, for a total of 99 individuals, giving a total of 1386 images. The image dimensions are 320 x 280, JPEG format [28][29][30].

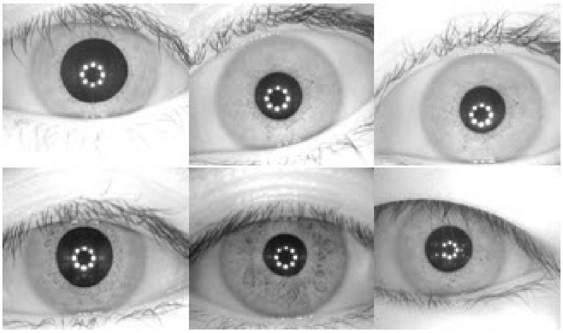


Fig. 2. Examples of the human iris images from the CASIA database

4 Proposed Modular Neural Network

The work was focused on the recognition of persons using a modular neural network with type-2 fuzzy integration, and image preprocessing methods to obtain the interest region of the iris.

4.1 Iris Image Pre-processing

The pre-processing that has been applied to the images before they are introduced to the modular neural network is as follows:

- Obtain the coordinates and radius of the iris and pupil using the method developed by Masek and Kovesi [31].
- Making the cut in the Iris.
- Resize the cut of the Iris to 21-21
- Convert images from vector to matrix
- Normalize the images.

4.2 Modular Neural Network with Type-2 Fuzzy Logic Integration

The proposed modular neural network consist of three modules, in each module the inputs are the preprocessed iris images, of which the first 33 persons are the input in the first module, the next 33 persons in the second module and the last 33 persons in the third module, given a total of 99 persons (792 for training – 594 for test in total and 264 for training – 198 for test in each module).

All the modules consist of a neural network with an input layer (the preprocessed image iris), two hidden layers and one output layer (see Figure 3).

Each module has two sub modules with different architecture and learning algorithm.

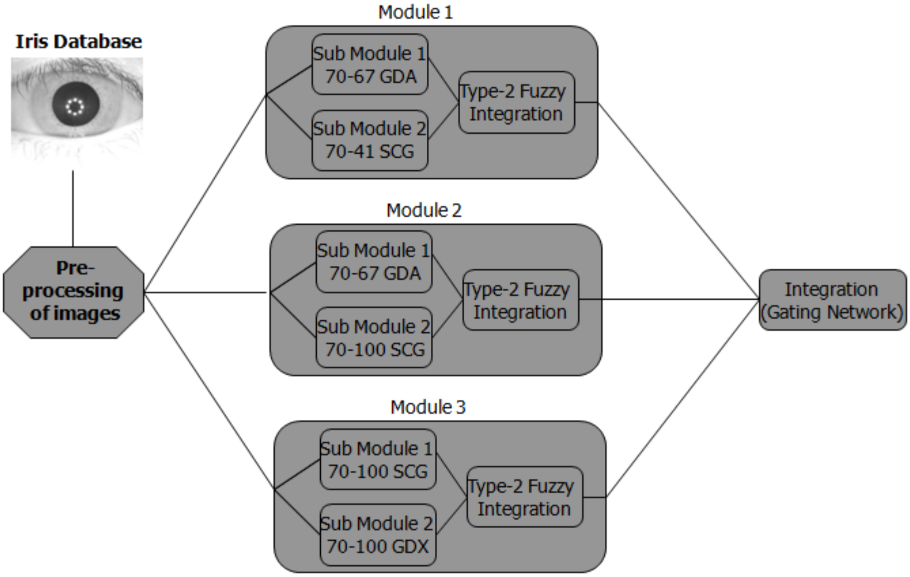


Fig. 3. Proposed modular neural network architecture

In the first module, the first sub module has 70 neurons in the first hidden layer and 67 neurons in the second hidden layer, and uses the gradient descent with learning rate algorithm for training; the second sub module has 70 neurons in first hidden layer and 41 neurons in the second layer, and uses the scaled gradient conjugate algorithm for training.

In the second module, the first sub module has 70 neurons in the first hidden layer and 67 neurons in the second hidden layer, and uses the gradient descent with learning rate algorithm for training; the second sub module has 70 neurons in first hidden layer and 100 neurons in the second layer, and uses the scaled gradient conjugate algorithm for training.

In the third module, the first sub module has 70 neurons in the first hidden layer and 100 neurons in the second hidden layer, and uses he scaled gradient conjugate algorithm for training; the second sub module has 70 neurons in first hidden layer and 100 neurons in the second layer, and uses gradient descent with momentum and learning rate algorithm for training.

The integration of the two sub modules in each module are realized with type-2 fuzzy integration, which has two inputs, the output of the first and second sub module (see Figure 4); the input has three membership functions, two of generalized bell type and one of trapezoid type; the output are the recognition person with two membership functions (see Figure 5 (a) and 5 (b)), one of trapezoid type and one of generalized bell type (see Figure 5 (c)); and work with nine rules (see Figure 6).

The integration of the modules is realized with gating network integration.

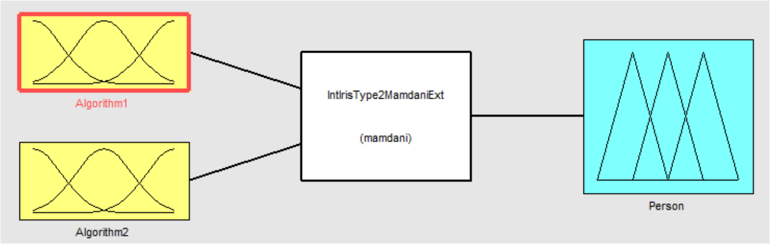
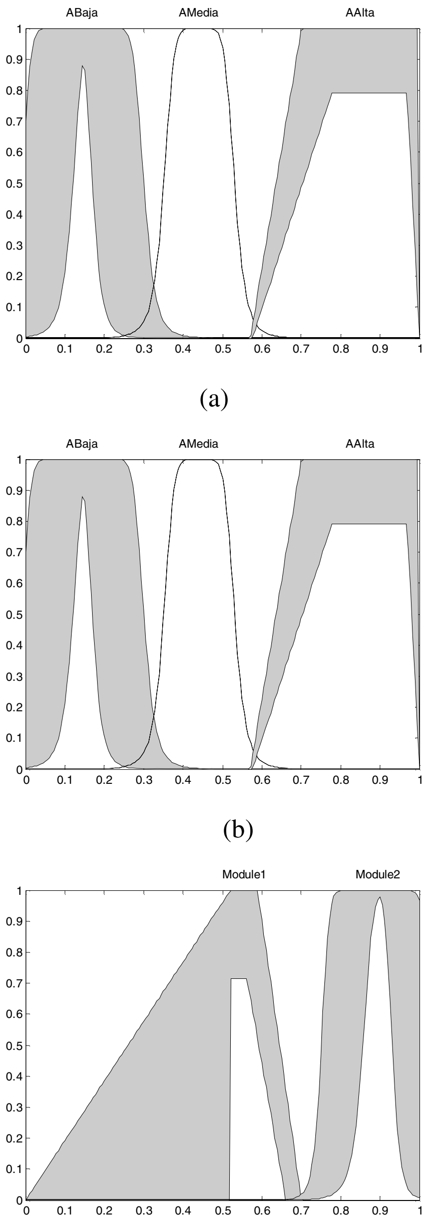


Fig. 4. Type-2 fuzzy integration system

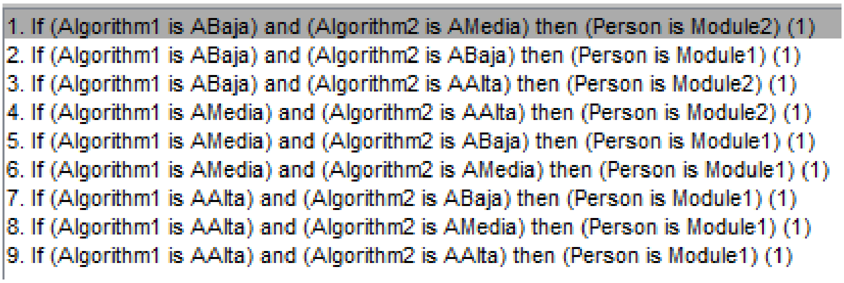


(a)

(b)

(c)

Fig. 5. (a) Membership functions of first input. (b) Membership functions of second input. (c) Membership functions of output.



1. If (Algorithm1 is ABaja) and (Algorithm2 is AMedia) then (Person is Module2) (1)
2. If (Algorithm1 is ABaja) and (Algorithm2 is ABaja) then (Person is Module1) (1)
3. If (Algorithm1 is ABaja) and (Algorithm2 is AAlta) then (Person is Module2) (1)
4. If (Algorithm1 is AMedia) and (Algorithm2 is AAlta) then (Person is Module2) (1)
5. If (Algorithm1 is AMedia) and (Algorithm2 is ABaja) then (Person is Module1) (1)
6. If (Algorithm1 is AMedia) and (Algorithm2 is AMedia) then (Person is Module1) (1)
7. If (Algorithm1 is AAlta) and (Algorithm2 is ABaja) then (Person is Module1) (1)
8. If (Algorithm1 is AAlta) and (Algorithm2 is AMedia) then (Person is Module1) (1)
9. If (Algorithm1 is AAlta) and (Algorithm2 is AAlta) then (Person is Module1) (1)

Fig. 6. Rules of the type-2 fuzzy integrator

5 Simulation Results

10 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	Total Rec.
T1	8000	0.000001	97.98 (582/594)
T2	8000	0.000001	96.29 (572/594)
T3	8000	0.000001	96.96 (576/594)
T4	8000	0.000001	96.13 (571/594)
T5	8000	0.000001	96.80 (575/594)
T6	8000	0.000001	96.29 (572/594)
T7	8000	0.000001	96.13 (571/594)
T8	8000	0.000001	95.95 (570/594)
T9	8000	0.000001	96.29 (/572594)
T10	8000	0.000001	95.95 (570/594)

The best result is a total recognition of 582 out of 594 images of iris of 99 persons; giving a recognition rate of 97.98 %.

The average of the 10 tests is 96.47 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	97.98 %
Gaxiola [32]	97.13 %
Sanchez-Avila [33]	97.89 %
Ma [28]	98.00 %
Tisse [18]	89.37 %
Daugman [21]	99.90 %
Sarhan [24]	96.00 %

6 Conclusions

In this paper we presented a modular neural network architecture with type-2 fuzzy integration, which has as input the database of human iris images, three modules with two sub modules for each module for training. 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 integration provides optimal results for integration of a modular neural network, because the results obtained are on average of 96.47 percent of recognition and a best result of 97.98 percent of recognition.

The type-2 fuzzy integration manages uncertainty that allows to work with more complex pattern, which allows obtaining better results.

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. Advances in Soft computing 34, 603–618 (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. Soft Computing for Hybrid Intelligent Systems, 171–186 (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] Morcego, B., Cembrano, G., Fuertes, J.: MIGA, A Software Tool for Nonlinear System Modelling with Modular Neural Networks. Applied Intelligence. Kluwer Academic Publishers (2004)

[7] Castillo, O., Melin, P.: Soft Computing for Control of Non-Linear Dynamical Systems. Springer, Heidelberg (2001)

[8] Zadeh, L.A.: Fuzzy Sets. Journal of Information and Control 8, 338–353 (1965)

[9] Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing, pp. 2–3. Springer, Heidelberg (2005)

[10] 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)

[11] 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)

[12] Jang, J., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice Hall, New Jersey (1997)

[13] Castillo, O., Melin, P.: Type-2 Fuzzy Logic Theory and Applications, pp. 29–43. Springer, Berlin (2008)

[14] Castro, J.R., Castillo, O., Melin, P.: An Interval Type-2 Fuzzy Logic Toolbox for Control Applications. FUZZ-IEEE, 1–6 (2007)

[15] 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)

[16] 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. Soft Computing for Hybrid Intelligent Systems, 89–114 (2008)

[17] 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)

[18] 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)

[19] Tisse, C., Martin, L., Torres, L., Robert, M.: Person identification technique using human iris recognition. Universite de Montepellier (2000)

[20] López, J., González, J.: State of the Art: Automatic Recognition of Human Iris. Politécnico Colombiano, and National University of Colombia, Scientia et Technica Año XI (29), 77–81 (2005)

[21] Khaw, P.: Iris recognition technology for improved authentication, Sala de Lectura de Seguridad de la Información, SANS Institute, pp. 1–17 (2002)

[22] 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)

[23] 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)

[24] 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)

[25] Sarhan, A.: Iris Recognition using Discrete Cosine Transform and Artificial Neural Networks. Journal of Computer Science 5, 369–373 (2009)

[26] 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)

[27] 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)

[28] Muron, A., Pospisil, J.: The human iris structure and its usages, Czech Republic, Physica, pp. 89–95 (2000)

[29] 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)

[30] 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

[31] 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)

[32] 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)

[33] Sanchez-Avila, C., Sanchez-Reillo, R., de Martin-Roche, D.: Iris Recognition for Biometric Identification using Dyadic Wavelet Transform Zero-Crossing. In: Proceedings of the IEEE 35th International, Camahan Conference on Security Technology, pp. 272–277 (2001)

Wavelet Neural Network Algorithms with Applications in Approximation Signals

Carlos Roberto Domínguez Mayorga[1], María Angélica Espejel Rivera[2],
Luis Enrique Ramos Velasco[3,4], Julio Cesar Ramos Fernández[3],
and Enrique Escamilla Hernández[4]

[1] Universidad Politécnica Metropolitana de Hidalgo. Camerino Mendoza no. 318 Col. Morelos
Pachuca, Hgo. C.P. 42040
[2] Universidad la Salle Pachuca, Campus La Concepción, Av. San Juan Bautista de La Salle
No. 1. San Juan Tilcuautla, San Agustín Tlaxiaca, Hgo. C.P. 42160. Pachuca, Hidalgo. México
[3] Centro de Investigación en Tecnologías de Información y Sistemas,
Universidad Autónoma del Estado Hidalgo, Pachuca de Soto, Hidalgo, México, 42090
[4] Universidad Politécnica de Pachuca, Carretera Pachuca-Cd. Sahagún, Km. 20, Rancho Luna,
Ex-Hacienda de Sta. Bárbara, Municipio de Zempoala, Hidalgo, México
[5] SEPI-ESIME Cul. IPN, Av. Santa Ana 1000,
Col. San Francisco Culhuacan, Del. Coyoacan, México

Abstract. In this paper we present algorithms which are adaptive and based on neural networks and wavelet series to build wavenets function approximators. Results are shown in numerical simulation of two wavenets approximators architectures: the first is based on a wavenet for approach the signals under study where the parameters of the neural network are adjusted online, the other uses a scheme approximators with an IIR filter in the output of wavenet, which helps to reduce convergence time to a minimum time desired.

1 Introduction

The contribution of computer science in the development of new and improved algorithms for approximation of signals in different areas of engineering, is now a reality [1], [2], [3]. Thanks to these contributions is now possible to consider the implementation of computational algorithms increasingly complex, as is the case of algorithms making use of neural networks and fuzzy logic as shown in [4,5,6,7,8,9].

In [10] it is use of wavelet neural networks for approximation of functions, dividing the work into two parts: the first part proposes a three-layer network (1, N, 1), where neurons of the input and output linear elements and the activation function of hidden layer is a mother wavelet, obtaining gradually the number of neurons (N) needed to cover the region of time and frequency of an objective function, while the translation and scaling parameters of the wavelet, and the weights of each neuron using the Kohonen rule are calculated, in the second part the error is minimized using the backpropagation algorithm. In addition, a comparison of results against those produced by a traditional backpropagation network are made, showing a faster convergence and better results for both training data and test data.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 374–385, 2011.

One notable difference between the algorithm proposed in [10] and the algorithm proposed in this paper is that the first one has an slower convergence, due to it must first be obtained sufficient numbers of neurons to cover the region of the objective function and then begins with minimizing the error, while the algorithm proposed here has an IIR filter to the output of wavelet network which serves as the optimization of the network (in this case discriminates neurons rather than adding them) during the minimization process.

In [11] a wavelet neural network robust based on the theory of robust regression is applied in the context of approximation of functions, adaptively adjusting the number of training data involved during training. To improve the robustness of wavenets networks, the training procedure for the initial parameters is carried out by the LTS (least square trimmer) algorithm described in [12]. In this case are two examples of approximation, one with 1-dimension function and the other with two dimensions. In contrast to the LTS algorithm proposed in [11] for network training, the one that is implemented in this article (LMS) does not require a preliminary manipulation of the errors (the LTS algorithm performs an array of errors before start), what could delay the operation of the algorithm.

In [13] is propose a method to implement the analog to digital converter (ADC) with high accuracy, using a wavelet neural network to approximate and compensate the nonlinearity of the ADC. The proposed network is a 3 layers network, where the output layer implements the sigmoid function, while the Morlet mother wavelet is implemented in the rest of the network. This type of network requires a small number of iterations and parameters compared with multilayer perceptrons. The algorithm proposed in [13] is very similar to this article, only that the output in this case is the identity function and in [13] is the function sigmoid, which delimits the output in the range $(0, 1)$, which show that is useful for the case being studied. Another difference with the algorithm here proposed is that IIR filter to the network output is not implemented.

In [14] a wavelet neural network is proposed for online identification problem, online identification, proposing an identification scheme based on wavelet neural networks and learning algorithm for online identification of nonlinear systems. This presents some techniques that could be implemented in future work to extend the algorithm proposed in this paper to problems in real time.

In [15] it is present fundamentals of networks wavenets and some of its applications. Are presented recurrent learning scheme in wavelet networks dynamics (which is very similar to that proposed in this article without the IIR in the output of the network), this type of learning has good results for numerical simulation in the approximation of functions. A comparison is also made with respect to the radial basis neural networks (which are already good approximations of functions) have several advantages over them. One of the most important application that is described in [15] which is predicting chaotic time series, for example, the Ikeda attractor, the Lorenz attractor. It also presents different variants of the learning techniques employed. In [15] are presented the basic theory of the wavenets that are implemented in this article, as well as some variants of the training algorithm implemented here, which was very useful in developing it.

The paper is organized as follows: in Section 2 the approximation of signals by wavenet is presented. In Section 3 the two wavenet architectures used in this article are studied, in Section 4 the numerical simulations results are presented. The comparison between the two architectures are presented in Section 5. Finally, conclusions about the results are presented in Section 6.

2 Wavelet Neural Network (Wavenets)

Combining the theory of wavelet transform with the basic concept neural network, we propose a new mapping network called adaptive wavelet neural network or wavenets as an alternative to neural networks to approximate an arbitrary nonlinear functions [16]. Wavenet algorithms basically consist of two processes: self-built networks and minimizing the approximation error. In the first trial, the network structures of representation are applied to specific analysis using wavenet. The network combines hidden units gradually to cover sufficiently efficient and time-frequency region occupied by a given goal. Simultaneously, network parameters are updated to keep the network topology and use the post process. In the second process, approximations of the instantaneous errors are minimized using an adaptive parametric technique based on gradient descent algorithm, i.e. initialized network parameters are updated using this method. Each hidden unit has a square window in the time-frequency plane. The rule of optimization is only applied to the hidden units where the selected point falls in their windows. Therefore, the cost of learning can be reduced. All these advantages have wavenets networks are exploited in approximately wavenet as used in this research.

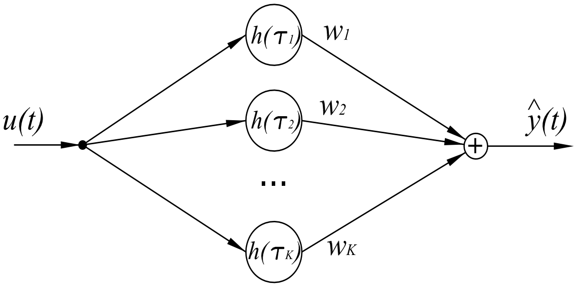


Fig. 1. Structure of wavelet network of three layers, where $\tau_k = \frac{t-b_k}{a_k}$, is the k-esima daughter argument, and $k = 1, 2, ..., K$

The architecture of wavenets shown in Figure 1 approximates the desired signal $u(t)$ by generalizing a linear combination of a set of wavelets daughters $h_{a,b}(t)$, where they are generated by a dilation a and b of translating the mother wavelet $h(t)$ [17], [18], [19], [20], [21]:

$$h_{a,b}(t) = h\left(\frac{t-b}{a}\right) \tag{1}$$

the dilation factor $a \neq 0, b \in \mathbb{R}$.

The approximation signal from the neural network $\hat{y}(t)$ can be represented by:

$$\hat{y}(t) = u(t) \sum_{k=1}^{K} w_k h_{a_k,b_k}(t) \tag{2}$$

where K is the number of wavelets windows and w_k are the weights and $h(t)$ is a mother wavelet, a_k and b_k are scaling and translation respectively of k-th neuron.

3 Wavenets Algorithms

3.1 Wavenet

To calculate the gradients used in the parameters update rules the energy function is defined as:

$$E = \frac{1}{2} \sum_{t=1}^{T} e^2(t) \tag{3}$$

where $e(t)$ represents the approximation error with respect to an objective function $u(t)$ and the output network $\hat{y}(t)$ is given as:

$$e(t) = u(t) - \hat{y}(t) \tag{4}$$

The objective is to minimize the function $E(w_k, a_k, b_k)$, varying the parameters w_k, a_k and b_k, where $k = 1, 2, \ldots, K$. For it we calculate the gradients:

$$\frac{\partial E}{\partial w_k} = - \sum_{t=1}^{T} e(t) u(t) h(\tau_k) \tag{5}$$

$$\frac{\partial E}{\partial b_k} = - \sum_{t=1}^{T} e(t) u(t) w_k \frac{\partial h(\tau_k)}{\partial b_k} \tag{6}$$

$$\frac{\partial E}{\partial a_k} = - \sum_{t=1}^{T} e(t) u(t) w_k \tau_k \frac{\partial h(\tau_k)}{\partial b_k} = \tau_k \frac{\partial E}{\partial b_k} \tag{7}$$

The increases in each coefficient are the negatives of their gradients,

$$\Delta \boldsymbol{w} = -\frac{\partial E}{\partial \boldsymbol{w}}, \quad \Delta \boldsymbol{a} = -\frac{\partial E}{\partial \boldsymbol{a}}, \quad \Delta \boldsymbol{b} = -\frac{\partial E}{\partial \boldsymbol{b}} \tag{8}$$

Thus the coefficients $\boldsymbol{w}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ of the wavenet are updated according to the rules

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \mu_{\boldsymbol{w}} \Delta \boldsymbol{w} \tag{9}$$

$$\boldsymbol{a}(t+1) = \boldsymbol{a}(t) + \mu_{\boldsymbol{a}} \Delta \boldsymbol{a} \tag{10}$$

$$\boldsymbol{b}(t+1) = \boldsymbol{b}(t) + \mu_{\boldsymbol{b}} \Delta \boldsymbol{b} \tag{11}$$

where μ is a fixed parameter that helps to improve the speed of learning of the wavenet, which is determined by trial and error.

Summarized the final algorithm which is given as:

Algorithm 1. The algorithm from wavenet is:

1. Compute the outputs of the wavenet $\hat{y}(t)$ as in (2) for $t = 1, 2, ..., T$, i.e., one iteration (epoch).
2. For each of the values in t to calculate the error $e(t)$ with respect to the input $u(t)$ defined in (4).
3. Obtain the error energy function E defined in (3) and compute $\frac{\partial E}{\partial w_k}$, $\frac{\partial E}{\partial a_k}$ and $\frac{\partial E}{\partial b_k}$ given in (5), (6) y (7), respectively.
4. Define the increments $\Delta \boldsymbol{w}$, $\Delta \boldsymbol{a}$ and $\Delta \boldsymbol{b}$ for the parameters $\boldsymbol{w}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ as (8).
5. Updates are performed of the parameters $\boldsymbol{w}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ according to (9), (10) and (11), respectively.
6. Repeat the number of iterations (epochs) necessary where the error is minimized or reaches some threshold while $\varepsilon > 0$.

3.2 Wavenets with Block IIR Structure

As mentioned above, a wavenet is a local network where the output function is well localized in both time domain and frequency. In addition, a local network can be achieved by combining two architecture wavenet in cascade with an infinite impulse response filter (IIR) [22]. The IIR recurrent loop creates a local structure that provides a method computationally efficient network training as a result is less time for convergence in the approximation of the signal. Figure 2 shows the structure of the network to approximate a signal $u(t)$, by generalizing a linear combination of a set of wavelets daughters, $h_{a,b}(t)$ arranged in cascade with the IIR filter. The signal is approximated by the network, $\hat{y}(t)$ is modeled by:

$$\hat{y}(t) = \sum_{i=0}^{M} c_i z(t-i) u(t) + \sum_{j=1}^{N} d_j \hat{y}(t-j) v(t) \tag{12}$$

where

$$z(t) = \sum_{k=1}^{K} w_k h_{a_k, b_k}(t) \tag{13}$$

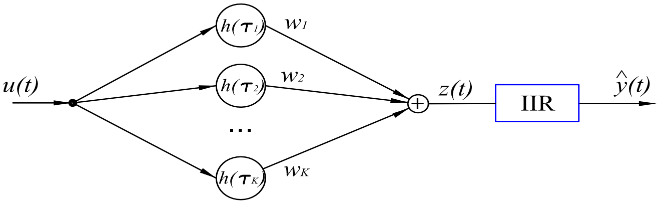


Fig. 2. Structure of wavelet network with IIR filter

K is the number of wavelets, w_k is the k-th weight, M is the number of feedforward coefficients c_i of the IIR filter, while N is the number of feedback coefficients d_j of the filter. The signal $u(t)$ is the input to be approximated and $v(t)$ is a persistent signal.

The wavenet-IIR parameters to be updated are w_k a_k, b_k, c_i and d_j, which are optimized using the gradient descent algorithm minimizing the error energy function, E defined as in (3), in the instant of time t.

The gradients $\frac{\partial E}{\partial w_k}$, $\frac{\partial E}{\partial a_k}$ and $\frac{\partial E}{\partial b_k}$ are computed to identified the parameters of wavenet, and $\frac{\partial E}{\partial c_i}$ and $\frac{\partial E}{\partial d_j}$ coefficients of the IIR structure required for the minimization of E can be expressed as:

$$\frac{\partial E}{\partial w_k} = -\sum_{t=1}^{T} e(t)u(t)\sum_{i=0}^{M} c_i h(\tau_k - i) \tag{14}$$

$$\frac{\partial E}{\partial b_k} = -\sum_{t=1}^{T} e(t)u(t)\sum_{i=0}^{M} c_i w_k \frac{\partial h(\tau_k - i)}{\partial b_k} \tag{15}$$

$$\begin{aligned}\frac{\partial E}{\partial a_k} &= -\sum_{t=1}^{T} e(t)u(t)\sum_{i=0}^{M} c_i w_k \tau_k \frac{\partial h(\tau_k - i)}{\partial b_k} \\ &= \tau_k \frac{\partial E}{\partial b_k}\end{aligned} \tag{16}$$

$$\frac{\partial E}{\partial c_i} = -\sum_{t=1}^{T} e(t)u(t)z(t-i) \tag{17}$$

$$\frac{\partial E}{\partial d_j} = -\sum_{t=1}^{T} e(t)v(t)\hat{y}(t-j) \tag{18}$$

Similarly, incremental changes of parameters are the negative of their gradients:

$$\Delta \boldsymbol{w} = -\frac{\partial E}{\partial \boldsymbol{w}},\ \Delta \boldsymbol{a} = -\frac{\partial E}{\partial \boldsymbol{a}},\ \Delta \boldsymbol{b} = -\frac{\partial E}{\partial \boldsymbol{b}},\ \Delta \boldsymbol{c} = -\frac{\partial E}{\partial \boldsymbol{c}},\ \Delta \boldsymbol{d} = -\frac{\partial E}{\partial \boldsymbol{d}} \tag{19}$$

thus, the vector of each coefficient $\boldsymbol{w}$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ wavenet network is updated by (9) (10) and (11), and for the IIR filter parameters using the rules:

$$\boldsymbol{c}(t+1) = \boldsymbol{c}(t) + \mu_{\boldsymbol{c}} \Delta \boldsymbol{c} \tag{20}$$

$$\boldsymbol{d}(t+1) = \boldsymbol{d}(t) + \mu_{\boldsymbol{d}} \Delta \boldsymbol{d} \tag{21}$$

where μ are fixed parameters that are determined by trial and error.

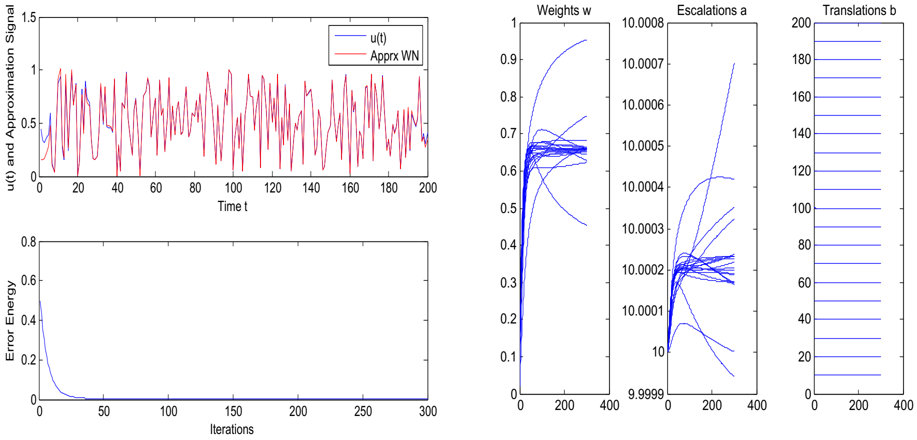
Algorithm 2. Wavenet-IIR algorithm that results is:

1. Compute the outputs of the wavenet-IIR $\hat{y}(t)$ as in (12) for $t = 1, 2, ..., T$, i.e., one iteration (epoch).
2. For each of the values in t to compute the error $e(t)$ with respect to input $u(t)$ defined in (4).

3. Obtain the error energy function E defined in (3) and compute $\frac{\partial E}{\partial w_k}$, $\frac{\partial E}{\partial a_k}$, $\frac{\partial E}{\partial b_k}$, $\frac{\partial E}{\partial c_i}$ and $\frac{\partial E}{\partial d_j}$ given by (14), (15), (16), (17) y (18), respectively.
4. Define the increments $\Delta \boldsymbol{w}$, $\Delta \boldsymbol{a}$, $\Delta \boldsymbol{b}$, $\Delta \boldsymbol{c}$ y $\Delta \boldsymbol{d}$ r the parameters $\boldsymbol{w}$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ as (19).
5. Updates are performed of the parameters $\boldsymbol{w}$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ y $\boldsymbol{d}$ according to (9), (10), (11), (20) and (21).
6. Repeat the number of iterations (epochs) required for the error is minimized or reaches a threshold while $\varepsilon > 0$.

4 Simulation Results

Then compare the behavior of the approximation wavenet with a bounded random signal generated by the function *random* given in MATLAB. For this, it is implemented a Morlet wavelet with $K = 20$, applying the algorithm to minimize the energy function of the error up to 400 iterations or an error threshold $\varepsilon = 0.001$.



(a)

(b)

Fig. 3. Wavenet approximation with 20 neurons and Morlet wavelet and its energy function (a). Update parameters during the learning process (b).

The Figure 4 shows the approximation Morlet wavenet with different number of neurons for example, we can see that for a number of neurons $K = 5$ in the wavenet takes 13 iterations to reach a threshold of error $e = 0.100$, while for the same threshold network with $K = 30$ requires 7 iterations and a wavenet with $K = 100$ the number of iterations required is 10. From this we can also see that the wavenet performing the approximation of the signal $u(t) = sin(t/20)$ was reached an error $e = 0.001$ with a smaller number of iterations has a number of neurons between $K = 60$ and $K = 100$ the same way we see that for a smaller number of neurons (about $K = 5$) requires a greater number of iterations to achieve rapprochement with minimum error values,

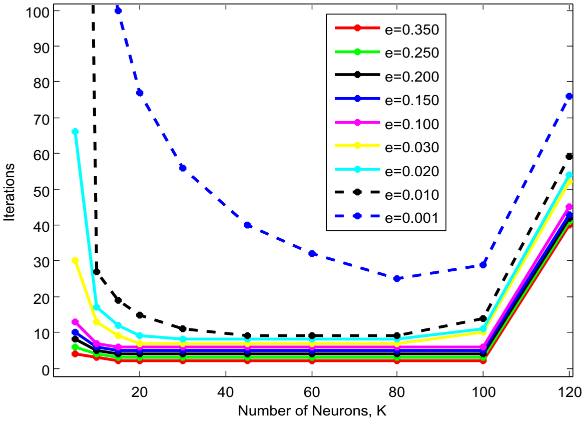


Fig. 4. Iterations required by the wavenets for different values of K with respect to different error thresholds

as in the case of a considerable amount of neurons (about $K = 100$), although errors in the range $0.1 \quad \leq e \leq \quad 0.5$ requires an approximate number of iterations for the wavenet different numbers of neurons, so for practical purposes compared wavenets different behaviors with $K = 20$, implemented only by changing the wavelet to the approximation of signals. We can conclude that the wavenet with better performance is the one with the wavelet Morlet as activation functions with 20 neurons, since the number of iterations required to approximate the signals under study were lower, also the processing time is reduced. Therefore below we use this wavenet Morlet for the approximation of noisy signals generated by the function *random* which is available in MATLAB.

4.1 Approach Morlet Wavenet-IIR and Different Number of Neurons

Table 1 contains some results of the approximation of a function implementing a wavenet-IIR in which change the number of neurons and obtained data on the number of iterations required to achieve some error values including the threshold. This is done only for values of K which have a good performance.

Table 1. Wavenets behavior for different values of K

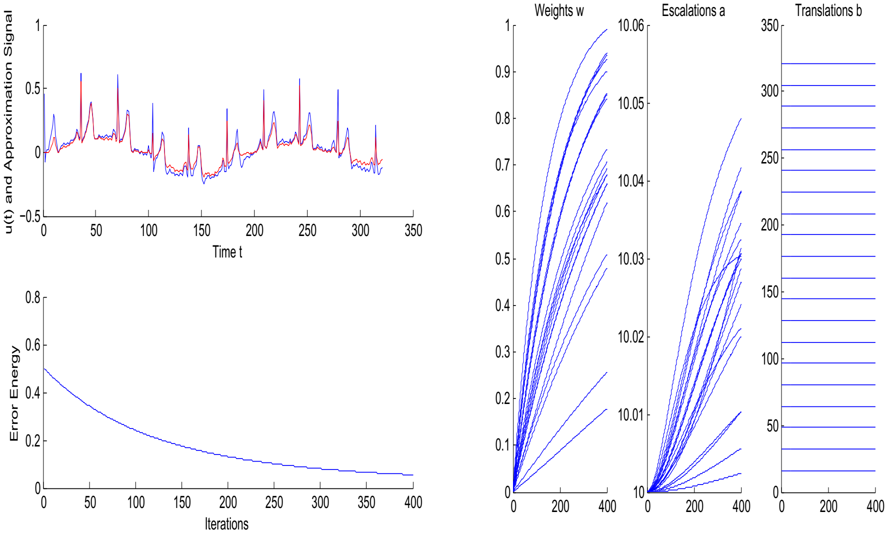
	Iteraciones									
$K \backslash e$	0.500	0.350	0.250	0.200	0.150	0.100	0.030	0.020	0.010	0.001
10	0	2	3	4	5	6	7	8	9	63
11	0	2	3	4	5	6	7	8	9	36
12	0	2	3	4	5	6	7	8	9	32
13	0	2	3	4	5	6	7	8	9	57

From this we can observe that with better wavenet-IIR behavior in the number of iterations required, is implementing 12 neurons, reaching the threshold at 32 iterations.

5 Comparison between Wavenet and Wavenet-IIR for ECG Signal Approximation

This section compares the performance between a wavenet and wavenet IIR, for this study a Morlet wavelet network is considered and an ECG signal, obtained from [23], is used.

In each case, making the approach to a maximum of 400 iterations or a threshold of minimum error of $\varepsilon = \ 0.001$, in a network of 20 neurons, the initial values parameters of the network are initialized the same values in both cases and for wavenet-IIR is used a number $M = 2$ and $N = 2$ for the IIR filter coefficients.

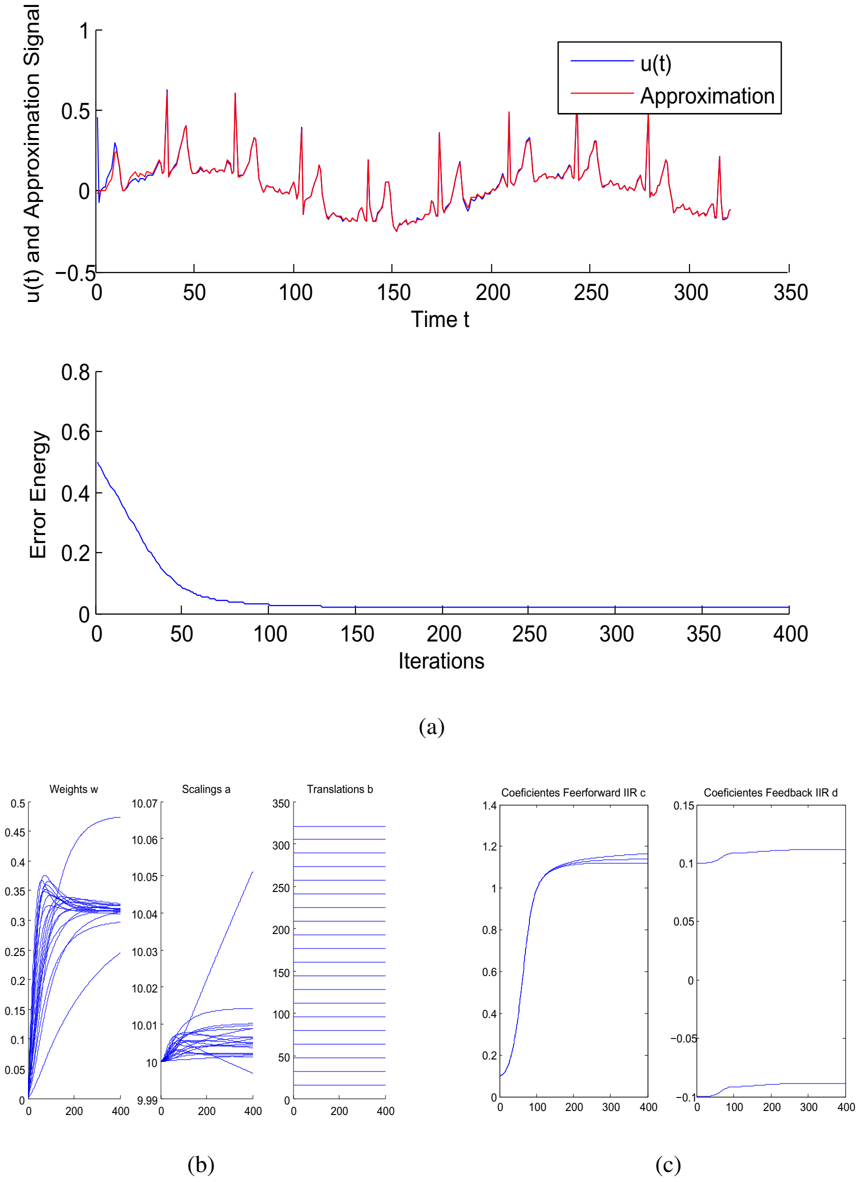


(a) (b)

Fig. 5. Wavenet approximation of an ECG and the energy of the error (a). Update the parameters of the network during the learning process (b).

Figure 5 shows the behavior of the wavenet approximation for ECG. It notes that the approach does not meet the threshold of the error before the 400 iterations and the speed with which the energy decreases the error is relatively slow.

Figure 6 shows the behavior of the wavenet-IIR approximation for ECG. It notes that the approach does not meet the threshold of the error before the 400 iterations, but the energy of the error decreases rapidly with respect to the approach that exists in wavenet.



(a)

(b)

(c)

Fig. 6. Wavenet-IIR approximation of the ECG signal and the error energy (a). Update the parameters of the network during the learning process (b) and IIR filter (c).

Table 2. Initial and final values of the wavenet parameters and wavenets-IIR shown in Figures 5 and 6, respectively

Initial					Final							
Wavenet y Wavenet-IIR					Wavenet			*Wavenet-IIR*				
w	***a***	***b***	***c***	***d***	***w***	***a***	***b***	***w***	***a***	***b***	***c***	***d***
0	10	16.1	0.1	0.1	1.1095	10.0013	16.0380	0.4300	10.7267	15.3363	1.2167	0.1084
0	10	32.2	0.1	−0.1	1.0243	10.0002	32.1056	0.2984	9.9128	32.2358	1.1390	0.1084
0	10	48.3	0.1	–	1.0562	10.0006	48.1451	0.3119	10.0613	48.0889	1.0766	–
0	10	64.4	–	–	1.0298	10.0005	64.2045	0.3079	10.0231	64.2390	–	–
0	10	80.5	–	–	1.0323	10.0004	80.2438	0.3019	10.0202	80.2118	–	–
0	10	96.6	–	–	0.5132	10.0002	96.3023	0.3080	10.0180	96.3326	–	–
0	10	112.7	–	–	1.0175	10.0007	112.3469	0.3088	10.0548	112.3075	–	–
0	10	128.8	–	–	1.0376	10.0005	128.4008	0.3063	10.0151	128.4059	–	–
0	10	144.9	–	–	0.9892	10.0005	144.4499	0.3112	10.0332	144.4573	–	–
0	10	161.0	–	–	1.0580	10.0006	160.4989	0.3059	10.0228	160.4780	–	–
0	10	177.1	–	–	0.9916	10.0004	176.5512	0.3085	10.0495	176.5980	–	–
0	10	193.2	–	–	0.4104	10.0001	192.5990	0.3027	10.0111	192.5760	–	–
0	10	209.3	–	–	1.0322	10.0006	208.6561	0.3033	10.0397	208.6869	–	–
0	10	225.4	–	–	0.9717	10.0005	224.6950	0.3105	10.0162	224.6748	–	–
0	10	241.5	–	–	01.0595	10.0007	240.7563	0.3024	10.0545	240.7889	–	–
0	10	257.6	–	–	0.9417	10.0003	256.7945	0.3089	10.0104	256.7712	–	–
0	10	273.7	–	–	0.8169	10.0003	272.8547	0.3087	10.0157	272.8775	–	–
0	10	289.8	–	–	0.9868	10.0007	288.8961	0.3044	10.0358	288.8820	–	–
0	10	305.9	–	–	0.9777	10.0005	304.9527	0.3041	10.0420	304.9681	–	–
0	10	322.0	–	–	0.8164	10.0002	320.9965	0.2886	10.0094	320.9788	–	–

6 Conclusions

The wavenets and wavenets-IIR is a good tool for signals approximation, showing a good performance. Although it is clear that the IIR structure gives a better performance with respect to the number of iterations required to achieve a fixed error threshold or simply minimize the energy function in the approximation of signals, random signals are bounded, algebraic signs (signs that represent algebraic functions) or medical signals (such as case of ECG).

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River (1994)
2. Haykin, S.: Kalman Filtering and Neural Networks. John Wiley & Sons, New York (2001)
3. Gupta, M.M., Jin, L., Homma, N.: Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory. John Wiley and Sons (2003)
4. Wang, J., Wang, F., Zhang, J., Zhang, J.: Intelligent controller using neural network. In: Yang, S.-Z., Zhou, J., Li, C.-G. (eds.) Proceedings SPIE Intelligent Manufacturing (1995)
5. Jun, W., Hong, P.: Constructing fuzzy wavelet network modeling. International Journal of Information Technology 11, 68–74 (2005)

6. Li, S.T., Chen, S.C.: Function approximation using robust wavelet neural networks. In: Proceedings 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002), pp. 483–488 (November 2002)
7. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation 3, 246–257 (1991)
8. Ting, W., Sugai, Y.: A wavelet neural network for the approximation of nonlinear multivariable function. In: IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC 1999 Conference Proceedings, vol. 3, pp. 378–383 (October 1999)
9. Wang, W., Lee, T., Liu, C., Wang, C.: Function approximation using fuzzy neural networks with robust learning algorithm. IEEE transactions on systems man and cybernetics Part B Cybernetics 27(4), 740–747 (1997)
10. Kobayashi, K., Torioka, T.: A wavelet neural network for function approximation and network optimization. In: Dagli, C.H., Fernandez, B.R., Ghosh, J., Soundar Kumara, R.T. (eds.) Proceedings of the Artificial Neural Networks in Engineering (ANNIE 1994) Conference on Intelligent Engineering Systems Through Artificial Neural Networks, vol. 4 (1994)
11. Li, S.T., Chen, S.C.: Function approximation using robust wavelet neural networks. In: 14th IEEE International Conference on Tools with Artificial Intelligence (2002)
12. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley (1987)
13. Chen, D.K., Han, H.Q.: Approaches to realize high precision analog-to-dogital comverter based on wavelet neural network. In: International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China (2007)
14. Gopinath, S., Kar, I., Bhatt, R.: Online system identification using wavelet neural networks. In: 2004 IEEE Region 10 Conference, TENCON 2004 (2004)
15. Sitharama, S., Cho, E.C., Phoha, V.V.: Fundations of Wavelet Networks and Applications. Chapman and Hall/CRC, USA (2002)
16. Zhang, Q., Benveniste, A.: Wavelet networks. IEEE Trans. Neural Networks (6) (November 1992)
17. Chui, C.K.: An Introduction to Wavelets. Academic Press Inc., Boston (1992)
18. Daubechies, I.: Ten lectures on waveletes. In: CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM (1992)
19. Mallat, S.: Wavelet Signal Processing. Academic Press (1995)
20. Teolis, A.: Computational Signal Processing with Wavelets. Birkhäuser, USA (1998)
21. Vetterli, M., Kovačević, J.: Wavelets and Subband Coding. Prentice-Hall, USA (1995)
22. Ye, X., Loh, N.K.: Dynamic system identification using recurrent radial basis function network. In: Proceedings of American Control Conference (1993)
23. Site, W.: (2009), http://www.physionet.org

Similar Image Recognition Inspired by Visual Cortex

Urszula Markowska-Kaczmar and Adam Puchalski

Wroclaw University of Technology, Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
urszula.markowska-kaczmar@pwr.wroc.pl

Abstract. The paper presents a method of image recognition, which is inspired by research in visual cortex. The architecture of our model called CaNN is similar to the one proposed in neocognitron, LeNet or HMAX networks. It is composed of many consecutive layers with various number of planes (receptive fields). Units in the corresponding positions of the planes in one layer receive input from the same region of the precedent layer. Each plane is sensitive to one pattern. The method assumes that the pattern recognition is based on edges, which are found in the input image using Canny detector. Then, the image is processed by the network. The novelty of our method lies in the way of information processing in each layer and an application of clustering module in the last layer where the patterns are recognized. The transformations performed by the CaNN model find the own representation of the training patterns. The method is evaluated in the experimental way. The results are presented.

Keywords: similar image recognition, visual cortex, shape analysis, Canny detector, neural network.

1 Introduction

Image recognition has always attracted interest of people. Probably it was the reason that it has became one of the research field in artificial intelligence that was founded on the claim that human intelligence can be simulated by a machine. Scientists aim to develop a method of image recognition comparable with human recognition. Therefore people look for inspiration in visual information processing in our brain. Though its detailed structure is still contentious among vision scientists and psychologists, they agree that visual cortex has a very active role in visual information recognition. Nature inspiration can offer us a real alternative in comparison to conventional approaches. This work describes a novel approach to similar image recognition based on computing that has been inspired by human visual information processing. Our network called CaNN is a modification of LeNet network [1]. Originality of this new model lies in the way of information processing in each layer and the last level of processing which is based on clustering.

The paper is organised as follows. The first part describes related works. Their short survey creates the background to show the proposed method. The

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 386–397, 2011.

next part shows the model in detail. It presents the idea, the structure and the method of training. Further part describes experiments that aim was to evaluate the model. The last section contains the future work and future direction of the system improvement.

2 Related Works

The foundations of the processing performed in the mammal visual cortex have been described in detail by David H. Hubel [2] and Torsten N. Wiesel [3] in the late 50's of the 20'th century. More recent information about modelling a visual processing in the brain of mammals can be found in [4], where a wide survey of various architectures and models of functions that may mediate perception are presented.

The idea presented by Hubel was inspiration for neural network community and it has been successfully implemented in the neural network called *neocognitron* [5] in the 80's by Fukushima. He creates a multilayered neural network mapping the structures discovered by Hubel and Wiesel onto layers of neural networks. Fukushima assumed, that the layers should be organised as a hierarchy of simple and complex cells appearing one after another. In *neocognitron* he managed to build 3 stages of this simple-complex cell combination. His first neocognitron was developed for handwritten digits recognition and was a great success. Then similar architectures were applied by others. These models are based on local receptive fields which provide an input to the units located next in the hierarchy. In the subsequent layers information is generalized and more and more complex patterns are recognized.

As it was mentioned, two basic cell types have been identified: simple cells (S) and complex cells (C). Simple cells (S) respond maximally to specific edge-like stimulus patterns within their receptive field. Complex cells (C) have larger receptive fields and are locally invariant to the exact position of the stimulus. A layer contains groups of units that are sensitive in many ways. A subset of these units with the same sensitiveness in a layer is called a feature map (or plane in the neocognitron). The feature maps in a layer have the same size and the same receptive fields.

Poggio and others [6] also use hierarchical architecture. They introduced a set of complex features combining position and scale tolerant edge detectors. Their system learns on the basis of a set with a small number of patterns. Our CaNN network described in the next section is a modification of convolutional network CNNs [1], called LeNet5. CNN exploits local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. The input hidden units in the i-th layer are connected to a local subset of units in the (i-1)-th layer, which are spatially contiguous.

Hierarchical networks have later been widely used in advanced character recognition systems [7] or image and object recognition systems [1], [6], [8], [9], [10].

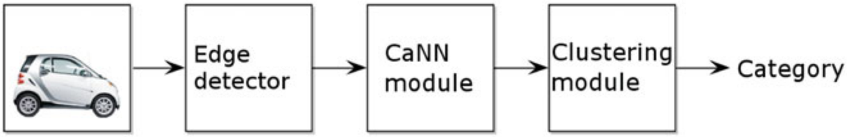


Fig. 1. Object recognition in the CaNN model

3 CaNN

Our CaNN model is dedicated to recognize a class (category) of object presented in the image in question. Information processing in the model is presented in Fig. 1. Input images are preprocessed by an edge detection system, which creates binary images including the edges contained in the image. The CaNN network transforms this image into its own representation expressed by a vector of values, where each value is in the range $< 0, 1 >$. This vector describes the input image as a set of complex features appearing somewhere in the image. On the basis of clustering module decision, the last stage determines a class of images the processed image belongs to.

The CaNN system has the similar structure to *CNN* network (*LeNet5*) [7]. The layer is composed of feature maps (planes). The first layer searches for simple features in an input image and the feature maps response giving information about recognized features by this layer. The next layer averages these features by position and scale. Feature maps search for complex features in the preceding layer, combining the previously separate elements together. The number of searched complex features is greater than the number of simple features in the first layer. The transformation performed in the next layers located in the hierarchy are similar to those already mentioned. The architecture of this model is important but the number of layers and feature maps in each layer has no significant meaning. The model ensures that the spatial resolution of the image is reduced while the number of feature maps in each layer is getting greater.

The last stage is a clustering module which decides which cluster the transformed image belongs to. The details of the described model are presented in the next sections.

3.1 Edge Detection

Many algorithms can be applied in the first stage. In our model after some initial experiments the Canny's edge detector [11] is used. Its implementation can be summarized in the following four stages:

1. *Noise reduction* – It is done on the basis of Gaussian filters.
2. *Gradient filtering* – Sobel operator in X and Y axis is applied. Then, the intensity and direction of the gradient is calculated.
3. *Non-Maximum Suppression* – This operation reduces the borders found in the gradient filtering to one pixel in width.

4. *Hysteresis filtering* – It creates final binary border image.

This preprocessing stage was introduced to prepare images for the later stages. It can be perceived as a model of the transformation that is performed by the Lateral Geniculate Nucleus in the mammalian brain. It is not considered in other networks like neocognitron [5], or LeNet [7] because these models work on binary images of characters.

3.2 CaNN Module

In this module on the basis of edges acquired in the edge detector simple (primitive) edge elements are built. Next, they are generalized to more complex structures and at the end CaNN finds the output vector being its own representation of the input image. The idea hidden in the CaNN processing ensures that complex input image represented by its binary map is step by step processed in the layers of CaNN.

The Architecture. The architecture of the CaNN neural network is presented in Fig. 2. As we mentioned, its architecture is similar to the networks already presented [5], [7]. Each stage in the CaNN structure is composed of two kinds of layers: Si – simple cells and Ci complex cells. Each layer contains a number of feature maps (or planes), illustrated in Fig. 2 by small rectangles. It is worth mentioning, that units (neurons) located in the same position, but in various feature maps of the same layer have the same receptive field located in the previous layer. Because they focus on the same receptive field they can be perceived as a column of planes sensitive to different features. These maps provide the response to the next layer in the hierarchy generalizing recognized features. The hierarchy ends with a feedforward neural network (two layered network, trained with Back-Propagation algorithm). The output layer of this network gives the final result of the whole CaNN network. The output vector is the internal representation of the processed input image.

Neural Network Training. The image processed by the Canny's filter creates an input for the whole network. The image is scaled to the dimensions 145 × 100 pixels. The edge detector does not change the dimensions of the image, so the picture has got the same width and height. Training of each layer is performed step by step and it will be described separately for each layer.

S1 layer. The receptive fields of neighboring neurons in one feature map of this layer get small part of image of the same size but shifted by 1 pixel. Each feature map is trained to be sensitive to one pattern representing eight border elements in different orientations (every $22,5^o$). The patterns are presented in Fig. 3. The aim of the *S1* layer is to detect simple edge fragments (primitives) in the input image. The layer is trained in a supervised way. The training pattern is a couple consisting of an image from Fig. 3 and its corresponding class is encoded as *1 of* n, where n is the number of primitives to be recognized. This number corresponds to the number of feature map (plane) in the *S1* layer.

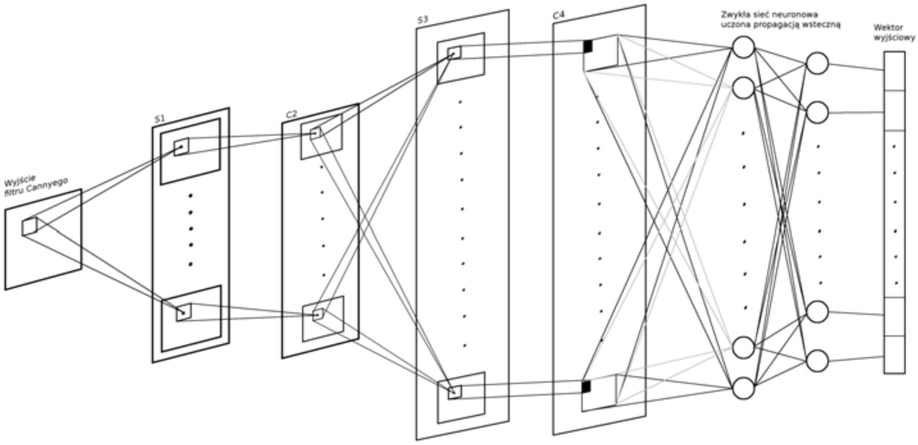
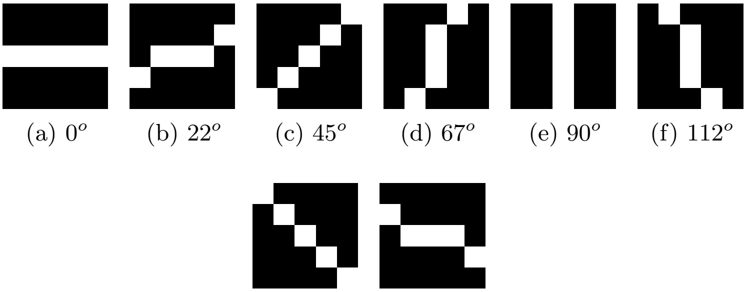


Fig. 2. Architecture of CaNN neural network

The edge extraction process is performed for every rectangular neighbourhood of pixels taken from the input image. The rectangle measures 5 × 5 pixels (Fig. 3) and the feature maps in the layer are of the dimensions 141 × 96 pixels.


(a) 0^o (b) 22^o (c) 45^o (d) 67^o (e) 90^o (f) 112^o

(g) 135^o (h) 157^o

Fig. 3. The training set for *S1* layer

Once pixels from an input image rectangle have been processed each feature map assigns its output, which can be 1 or 0. The neuron response of the concrete feature map can be described using the formula given by eq. 1.

$$p_{S1j}(\mathbf{n}) = (nn(\gamma(\mathbf{n}, \rho_{S1})))_j \tag{1}$$

where $\mathbf{n} = [x, y]$ is a neuron position in feature map, nn represents the transformation realised by the neural network in this layer. The function $\gamma(\mathbf{n}, \rho_{S1})$ transforms the given vector $\mathbf{n}$ and range ρ_{S1} returning a set of coordinates representing a rectangular neighbourhood of pairs (x_n, y_m). This rectangular field is limited with its left, upper corner $(\mathbf{n})$ and a range ρ_{S1} which in case of the

S1 layer has the value of 5×5. The *nn* receives a set of coordinates and using them it gets values from the input image (represented by edges detected by the Canny's edge detector). As a result, from all feature maps of this layer a vector of the values of 1 or 0 is obtained (1 means that the pattern has been recognized). The subscript $_j$ that appears in the formula means the j-th component of the vector, which is equivalent to j-th feature map in the layer. It can be noticed on the basis of the left side of the presented formula, which shows a concrete feature map of layer `S1` identified by j. The selected component from the output vector is placed in this j-th feature map in the coordinates indicated by the vector $\mathbf{n}$.

C2 layer. It transforms each result of the layer *S1* feature map averaging the values in a small neighbourhood (of 3×3 pixels) and put them in a concrete point in a specific feature map of the layer *C2*. The described neighbourhoods are, differently as it was in layer *S1*, non overlapping. The number of feature maps for both of the layers is equal to eight. After these two stages the feature maps (of dimensions 47×32) contain simple edge fragments which position is averaged and still the edge fragments are grouped according to the angle of the represented border. The average is calculated as presents the formula 2.

$$p_{C2j}(\mathbf{n}) = avg(p_{S1j}(\gamma(\mathbf{u}, \rho_{C2}))) \tag{2}$$

where: $\mathbf{n} = [x, y]$ has the same meaning like in *S1* layer, also γ function works in the same way as it was described, $\mathbf{u} = 3 \cdot \mathbf{n} = [3x, 3y]$, ρ_{C2} has the dimensions 3×3. The feature map p_{S1j} in the position $[x, y]$ gets this set of coordinates and returns a set of values that the j-th feature map holds in the provided arguments. The values got from the input maps p_{S1j} are averaged and produced as an output of feature map p_{C2j} in the coordinates shown by the vector $\mathbf{n}$.

S3 layer. This layer combines primitive features received from the precedent layer into greater parts of edges. The idea of processing in this layer is described in pseudocode by **Algorithm** 1. An average simultaneously calculated for various feature maps is the common transformation performed here. The feature maps in *S1* layer are sensitive to primitive patterns sorted with an increased angle of primitive edges. How it is organized in *S3* layer ? Feature map connectivity is presented in Table 1 and it can be easily interpreted. The rows refer to the concrete feature maps from the layer *C2*, which contain edge fragments of a specific angle. The columns are the feature maps representing the output of layer *S3*. In the table one can see that the first result feature map contains transformed edge fragments of the first three feature maps from the precedent layer. The resulting feature maps from this layer are of size 45×30 pixels because the rectangular neighbourhood of pixels is of dimensions 3×3. The layer contains 28 feature maps in the result set.

C4 layer. It is basically a clone of *C2*. The only difference are the parameters. The transformation and a whole logic of this layer is the same. The non overlapping neighbourhoods are of size 3×3. Therefore the feature maps have the dimensions of 15×10 pixels.

Algorithm 1 . Algorithm of the value $p_{S3j}(\mathbf{n})$ calculation

```
$cp \leftarrow connection\ pattern\ 1$
$C2 \leftarrow output\ from\ layer\ C2$
$j := feature\ map\ number$
$N \leftarrow \mathbf{n}$
$\rho \leftarrow \rho_{S3}$
$counter := 0$
$sum := 0$
for $k = 1 \rightarrow cp.rowCount$ do
    if $cp[k, j] = x$ then
        $w_{C2} \leftarrow C2[k]$ {choose a concrete feature map}
        for $[x, y] \in \gamma(N, \rho)$ do
            $sum+ = w_{C2}[x, y]$
            $counter+ = 1$
        end for
    end if
end for
$p_{S3j} := sum/counter$
return $p_{S3j}$
```

BackPropagation neural network. The last part of CaNN is created by two-layered neural network, trained with BackPropagation (BP). This network has a basic 2 layered fully interconnected architecture where the first layer contains 4200 neurons. In the basic version of the model the last layer contains 100 neurons. At the beginning the network is created with random weights. The training of this layer is supervised, so the network needs an input and a desired output. The input for this network is an output of precedent layer after processing an input image. The output is not chosen arbitrarily. The neural network with randomly chosen weights processes the input vector and the obtained result is then rounded up to 1 or 0. It creates a desired output for this network. Using this heuristics we can perceived the output vector as a hash code of an input image. The training process is stopped when the overall error is less than a specific treshold value chosen at the beginning.

Table 1. Connectivity patterns in *S3* layer

Input\output	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1 - 0^o	x						x	x	x					x	x	x	x				x	x		x	x			x
2 - $22,5^o$	x	x						x	x	x					x	x	x	x				x	x		x	x		
3 - 45^o	x	x	x						x	x	x					x		x	x				x	x		x	x	
4 - $67,5^o$		x	x	x					x	x	x	x					x		x	x				x			x	x
5 - 90^o			x	x	x					x	x	x	x				x	x		x	x				x			x
6 - $112,5^o$				x	x	x					x	x	x	x				x	x		x	x			x	x		
7 - 135^o					x	x	x					x	x	x	x				x	x		x	x			x	x	
8 - $157,5^o$						x	x	x					x	x	x	x				x	x		x	x			x	x

3.3 Clustering Module

The calculations performed in the precedent layers of the network transform an image into a vector of real values from the range $< 0, 1 >$. Considering a huge image set processed by the precedent layers of the network, a set of vectors being an internal neural network representation of these images are received as outputs of CaNN. In the next step these vectors are clustered. The clusters will be represented by the mean vector calculated on the basis of representative vectors obtained from the last layer for the given input image class. This mean vector can be considered as a center of a cluster that represents images from the concrete category (input image class). Once the module is trained it is ready to use. Though because the assignment of image category is based on the searching of the nearest center of cluster it is necessary to assume a distance metrics. For simplicity in the first experiments Euclidian distance was taken. The clustering process takes an advantage of knowledge we have about the input image classes, therefore the training is very fast.

4 Experimental Study

The main objective of the experimental study is a proof of concept demonstrating the feasibility of CaNN and its ability to recognize image category. The model has been based on many heuristics and it would be difficult to prove its efficiency analytically. Therefore it will be evaluated in an experimental way. Our intention was to plan the experiments in such way that they show advantages and drawbacks of the proposed model. It would be desired to compare the presented model with others like neocognitron CNN and HMAX, but the first two networks were developed to the specific domain – to recognize letters and digits, HMAX has been patented, so difficult available, therefore at this moment there is no comparison with other similar networks. Instead, we have focused on investigation of the network sensitivity to parameters and on evaluation of the final recognition.

4.1 Training and Testing Data

We have collected 343 images in training set and 123 in the testing one. All images belong to 18 input classes (categories), namely: shoes, cars, old postcards of the city (that represent 13 classes), images of *Notre Dame* cathedral, images of the *Mona Lisa* painted by Leonardo da Vinci and the *Scream* painted by Munch. The rationale to such a choice was as follows. The first two classes were chosen because of the shape similarity. The postcards present the same places in the city but in the stretch of 80 years. *Mona Lisa* and *Scream* have many modified images in the Internet. To create a challenge for the model the test set has the same classes of images but the images differ much in comparison to the training set. An exemplary small subset of images used in this project is presented in Fig. 4. The training set is shown in Fig. 4a), the testing set in Fig. 4b).

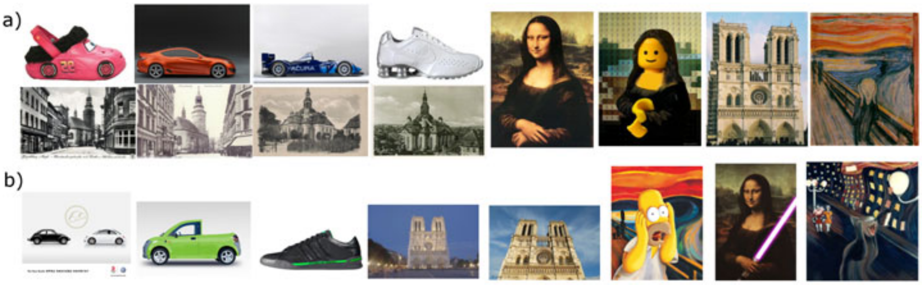


Fig. 4. Exemplary patterns: a) – from training set and b) – testing sets

4.2 Experiment 1 – The Pattern Recognition Related to the Number of the CANN Outputs

The first experiment has been performed to investigate the influence of the output vector size (the number of outputs in the output layer in the two layered network) of CaNN network on the final classification result measured by wrong matching. The whole model has been trained using the described above training set and tested on the test set. In the final clustering the Euclidian distance has been used. The results of this experiment are shown in Fig. 5.

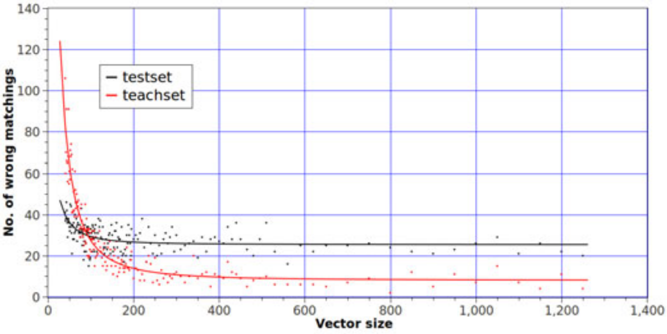


Fig. 5. The influence of the output vector size on the results of classification (Experiment1)

One would have expected that a small increase in the vector size would increase the accuracy of the final classification but it is not observable in Fig. 5. To make the trend more visible the results are approximated with a homographic function. As it can be noticed, the model is not sensitive to the change of this parameter. It remains stable and the results are not getting worse in any way.

4.3 Experiment 2 – The Influence of the Assumed Training Error on the Pattern Recognition

In this experiment the sensitivity of the model on the assumed training error is tested. How much this parameter influences on the result can be seen in the figure 6. The results have been gathered for 30 trials of the model creation with the same parameters (with the neural networks created randomly). In Fig. 6 the result values are approximated by the second order polynomial.

The result of this experiment has showed that a very small training error not always gives good results. Of course, the results obtained with the training set are getting better when this value is getting smaller. In the test set though, the trend seems to be different. One can see, that there exists a local minimum between of 0.8 and 0.9. This is the effect of test set creation. It includes many images varying much from training set.

4.4 Experiment 3 – The Influence of the Distance Metrics on the Pattern Recognition

In this experiment the influence of the assumed distance measure was tested. Euclidean distance is replaced by a more general formula - the Minkowski metrics (equation 3). As one can see the Minkowski distance can be transformed to the Euclidean metrics when the parameter $m = 2$. The results of this experiment are shown in Fig. 7.

$$L_m(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^{n} |x_i - y_i|^m)^{1/m} \tag{3}$$

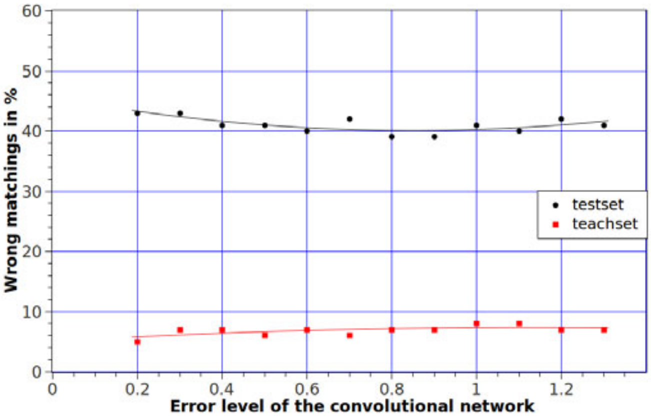


Fig. 6. The wrong matching in relation to the assumed training error – the result of the Experiment 2

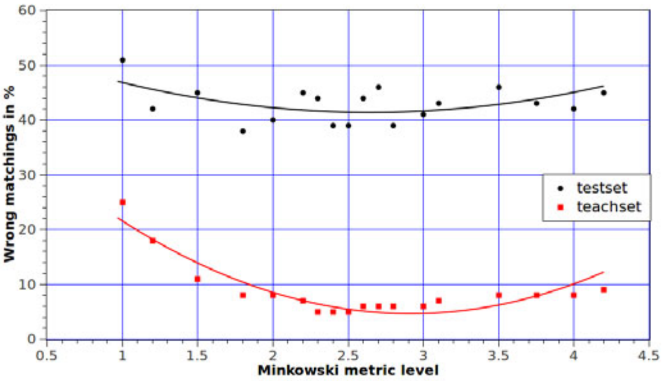


Fig. 7. The influence of the metrics applied in the clustering module – the results of Experiment3

The experiment has been performed also 30 times for each parameter combination (just like in the Experiment 2). The results have also been approximated by the parabolas. A minimum error exists (considering the measured range of Minkowski level) in the graph in Fig. 7. The best results have been obtained for levels 2.3, 2.4, 2.5 which makes the interpretation a bit difficult. Still, the result is worth mentioning.

5 Conclusion and Future Work

The model presented in this paper is significant modification of other networks based on visual cortex proposed so far. The introduced modifications refer to transformations performed in the network layers. Efficiency of the method was tested in the experimental way. In the presented experiments we have focused on investigation of the influence of some parameters on the final results. It seems that the obtained results allow to think positively about future development of the model.

We have also tested the ability of the model to produce the image on the basis of the representative vectors obtained as the output CaNN network (before clustering model). Though the results are difficult to show, this experiment has showed that it was possible to return some parts of input image on the basis of the vectors - being the model representations of the input images. This problem will be a subject of future research.

In further works another edge filter in the preprocessing stage will be considered (for instance Gabor filter). It is planned to enhance the model by adding two new layers *S5* and *C6*, as in other approaches based on visual cortex, to check the improvement of the network results.

Acknowledgment. This work was partially supported by the Innovative Economy Programme project POIG.01.01.02-14-013/09-00.

References

1. LeCun, Y., Kavukvuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Proc. International Symposium on Circuits and Systems. IEEE (2010)
2. Hubel, D.H.: Evolution of ideas on the primary visual cortex, 1955-1978: A biased historical account. Technical report, Harvard Medical School, Department of Neurobiology, Boston, Massachusetts, U.S.A (1981)
3. Wiesel, T.N.: The postnatal development of the visual cortex and the influence of environment. Bioscience Reports 2, 351–377 (1982)
4. Ng, J., Bharath, A., Zhaoping, L.: A survey of architecture and function of the primary visual cortex (v1). EURASIP Journal on Advances in Signal Processing, 1–17 (2006)
5. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 193–202 (1980)
6. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 411–426 (2007)
7. Yann, L., Leon, B., Yoshua, B., Patrick, H.: Gradient-Based Learning Applied to Document Recognition. In: Proceedings of the IEEE Workshop, pp. 1–7. IEEE Computer Society (1998)
8. Riesenhuber, M.: How a Part of the Brain Might or Might Not Work: A New Hierarchical Model of Object Recognition. PhD thesis. MIT (2000)
9. Serre, T., Poggio, T., Wolf, L.: Object Recognition with Features Inspired by Visual Cortex. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 994–1000 (2005)
10. Serre, T., Poggio, T., Riesenhuber, M., Wolf, L., Bileschi, S.M.: High-Performance Vision System Exploiting Key Features Of Visual Cortex. Technical report, Massachusetts Institute of Technology, United States Patent Application Publication (2006)
11. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (6), 679–698

Regularization with Adaptive Neighborhood Condition for Image Denoising

Felix Calderon and Carlos A. Júnez–Ferreira

Universidad Michoacana de San Nicolás de Hidalgo
División de Estudios de Posgrado.
Facultad de Ingeniería Eléctrica
Santiago Tapia 403 Centro.
Morelia, Michoacán, México.
CP 58000
calderon@umich.mx, cjunez@www.fic.umich.mx

Abstract. Image denoising by minimizing a similarity of neighborhood-based cost function is presented. This cost function consists of two parts, one related to data fidelity and the other is a structure preserving smoothing term. The latter is controlled by a weight coefficient that measures the neighborhood similarity between two pixels and attaching an additional term penalizes it. Unlike most work in noise removal area, the weight of each pixel within the neighborhood is not defined by a Gaussian function. The obtained results show a good performance of our proposal, compared with some state-of-the-art algorithms.

Keywords: image denoising, regularization and neighborhood.

1 Introduction

Noise removal is an important task in image processing in terms of preparing an image for procedures such as segmentation or object recognition. Usually, observed data g in an image can be modeled by the expression:

$$g = f + \eta \tag{1}$$

where f is the original image and η is additive noise. In order to suppress noise in images, several techniques have been proposed, among which we can find popular algorithms like the Bilateral filter [1] and Non-Local Means [2], both methods considered very effective for removing Gaussian noise. Nonlinear filters [3], usually based on median value [4], have been widely used in order to suppress impulsive noise. Another active area is image regularization based on minimization of functions. One of the early work on this issue was introduced by Rudin et al. in [5], where is proposed the Total Variation (TV) minimization problem:

$$\hat{f} = \arg\min_{f} \int_{\Omega} ||\nabla f|| \tag{2}$$

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 398–406, 2011.

subject to

$$||f - g||^2 = \sigma_\eta^2 \tag{3}$$

where $||\nabla f||$ is the Euclidean norm of image gradient. Constraint (3) considers the *a priori* information that noise standard deviation is σ_η.

In general, the estimated image $\hat{f}$ is computed by (4) according to [6]:

$$\hat{f} = \arg\min_{f} U(f) \tag{4}$$

where

$$U(f) = D(f) + \lambda R(f) \tag{5}$$

The data-fidelity term D establishes that the reconstruction f should be consistent with the observed data g. The regularization term R, usually, is the image gradient and promotes the formation of planar regions in the solution [7]. The contribution of the regularization term is controlled by parameter λ. In general when λ values is increased, details in the filtered image are lost. However, such methods do not exploit information provided by existing structures in images and could fail when gradients are introduced by the presence of high noise levels, such as impulse noise. In order to overcome this issue, this paper proposes a Regularization with Adaptive Neighborhood Condition (RANC) that performs a structure preserving smoothing controlled by non-Gaussian weights that depend on similarity between neighborhoods.

2 Functional Proposal

Chan et al., in [8], established that the data-fidelity and regularization terms of the objective function are given by (6) and (7):

$$D(f) = \sum_{r \in L} |f_r - g_r| \tag{6}$$

$$R(f) = \sum_{r \in L} \sum_{s \in W_r} |f_r - f_s| \tag{7}$$

where f_r and f_s are intensity values at positions r and s, respectively. $r = [i, j]^T$ represents a pixel position on the image lattice L, g_r is the observed intensity value at position r and the search neighborhood W_r is the set of all positions in the neighborhood centered at position r, i.e. considering a window size of $(2m + 1) \times (2m + 1)$, this set can be defined by

$$W_r = \{r + [k, l]^T | -m \leq k \leq m, -m \leq l \leq m\} \tag{8}$$

In general the window size will be smaller than the image size and it is accountable for the image smoothness degree according with (7). Short window size will produce sharp and clear images in contrast with large window size, which produces over–smoothed images.

As mentioned, this kind of regularization term promotes planar regions in restored images, suppressing noise effectively, but also details. Then, it is necessary to introduce spatial considerations in order to restore an image, through exploitation of the existence of repetitive structures in the image, idea introduced by Buades et al. in [2], known as Non-Local Means (NLM), where a pixel restoration is performed through the weighted average in a search neighborhood W_r:

$$\hat{f}_r = \sum_{s \in W_r} p_{r,s} g_s \tag{9}$$

The weights $p_{r,s}$, in equation (9), consider the similarity between neighborhoods of pixels at positions r and s as a function of the Euclidean distance given by (10)

$$d\left(V_r, V_s\right) = ||q\left(V_r\right) - q\left(V_s\right)||^2 \tag{10}$$

where $q(V_r)$ and $q(V_s)$ are vectors containing pixel intensity values of similarity neighborhoods V_r and V_s, respectively, such that,

$$\begin{array}{c} q(V_r) = [f(i-m, j-m), f(i-m, j-m+1)\dots \\ f(i-m, j+m), f(i-m+1, j-m), \dots \\ f(i+m, j+m)]^T \end{array} \tag{11}$$

if we consider a square neighborhood.

Thus, the weight between pixels at sites r and s was proposed as:

$$p_{r,s} = \frac{1}{Z_r} e^{-\frac{d(V_r, V_s)}{h^2}} \tag{12}$$

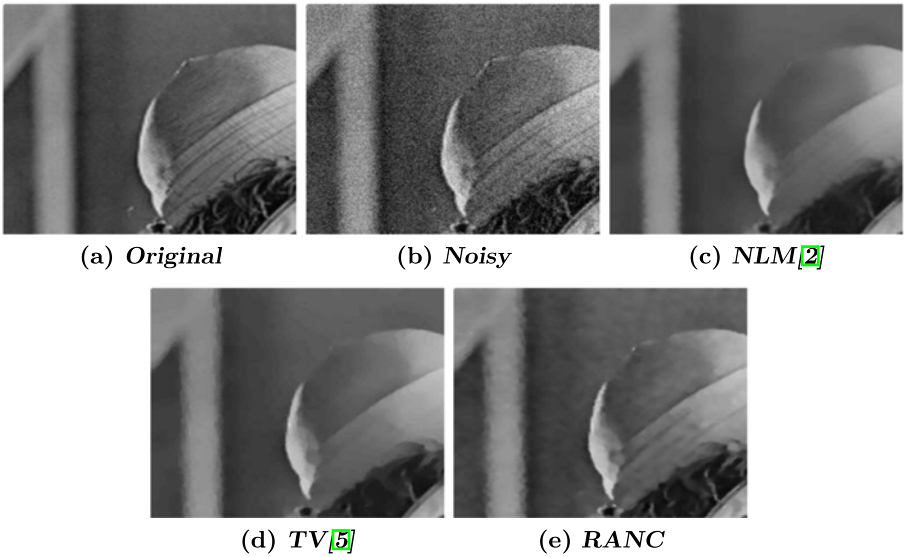
where Z_r is a normalizing factor such that $\sum_s p_{r,s} = 1$, and parameter h controls the decay of weights.

Incorporating the mentioned idea to image regularization, we propose the minimization of a function of the form as (5) with:

$$R\left(f, w\right) = \sum_{r \in L} \sum_{s \in W_r} \left[\left(1 - w_{r,s}\right)^2 d\left(V_r, V_s\right) + \mu w_{r,s}^2\right] \tag{13}$$

where $w_{r,s}$ acts as an adaptive weight to the contribution of f_s to the restoration. Parameter μ controls a penalty to this weight. At difference with (5) with regularization term as (7), where λ is constant, if we define for (13) $\widehat{\lambda}_{r,s} = \lambda(1-w_{r,s})^2$, the new regularization parameter $\widehat{\lambda}_{r,s}$ is variable giving an adaptive restoration which depends of the r-th and the s-th neighborhoods.

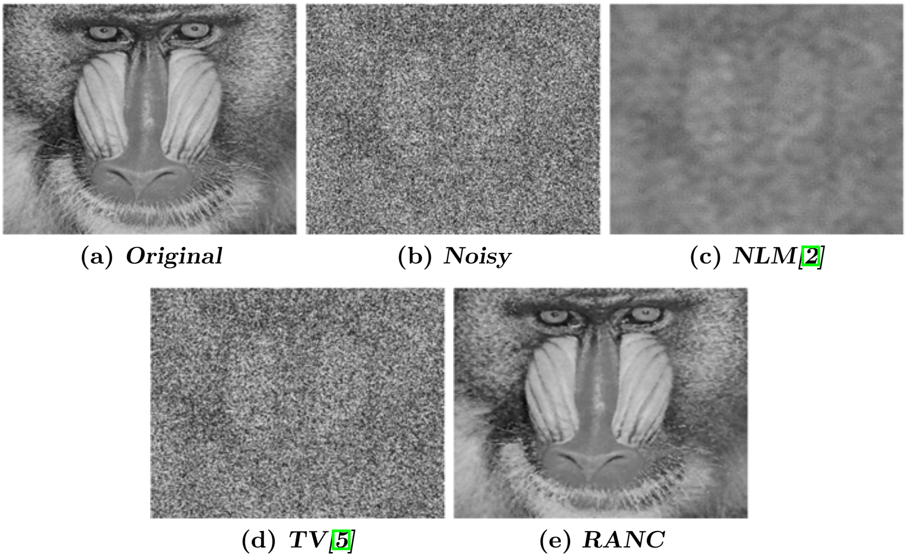
In the case of images with high levels of noise, as impulsive noise, the data-fidelity term given by (6), could provide a wrong influence to the restoration. In order to overcome this problem, the use of an intensity value most appropriate is proposed instead of the observed value g_r, i.e., an estimation $H(g_r)$ of the intensity value that probably the pixel at position r could have, obtained by filtering the observed image with the operator H, which can be linear or nonlinear. In the case of impulsive noise, this operator can be based on the intensity median



(a) ***Original*** (b) ***Noisy*** (c) ***NLM[2]***

(d) ***TV[5]*** (e) ***RANC***

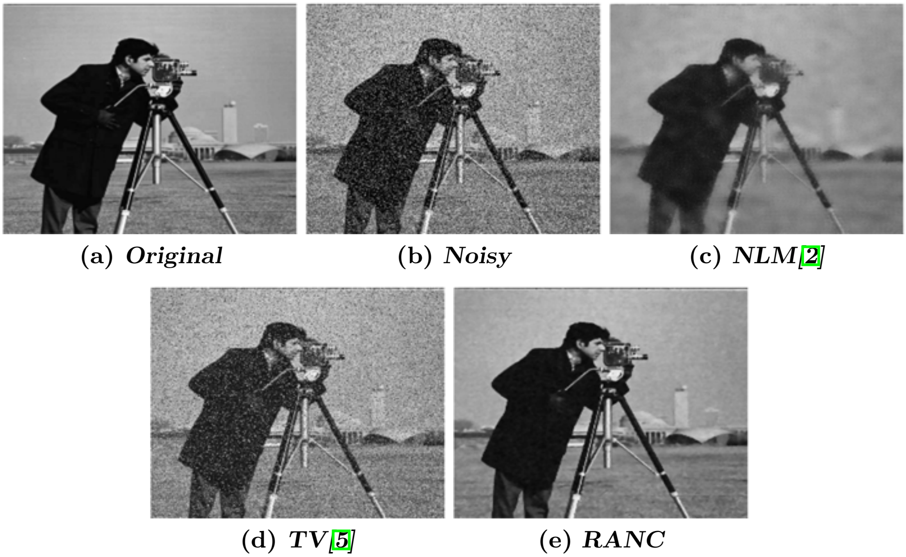
Fig. 1. Zoom of denoising experiment for Lena image with gaussian noise ($\sigma_\eta = 20$)



(a) ***Original*** (b) ***Noisy*** (c) ***NLM[2]***

(d) ***TV[5]*** (e) ***RANC***

Fig. 2. Denoising experiment for Baboon image with Salt and Pepper noise ($\delta = 80$)


(a) ***Original*** (b) ***Noisy*** (c) ***NLM[2]***

(d) ***TV[5]*** (e) ***RANC***

Fig. 3. Denoising experiment for Cameraman image with gaussian ($\sigma_\eta = 20$) and Salt and Pepper (20%) mixed noise

value of pixels belonging to the neighborhood around the pixel in question, due to its well-known robust outliers removal and edge preserving abilities [3]. Thus, the data-fidelity term can be expressed as:

$$D\left(f\right) = \sum_{r \in L} \left(f_r - H(g_r)\right)^2 \tag{14}$$

3 Implementation

The solution of the function given by (5) using (13) and (14) can be obtained considering the conditions $\frac{\partial U}{\partial w_{r,s}} = 0$ and $\frac{\partial U}{\partial f_r} = 0$. The obtained system can be solved using a Gauss-Seidel algorithm because the resulting linear system has a sparse, semi-positive and dominant diagonal values on the matrix. The Gauss-Seidel *(t)*-th iteration is given by equations:

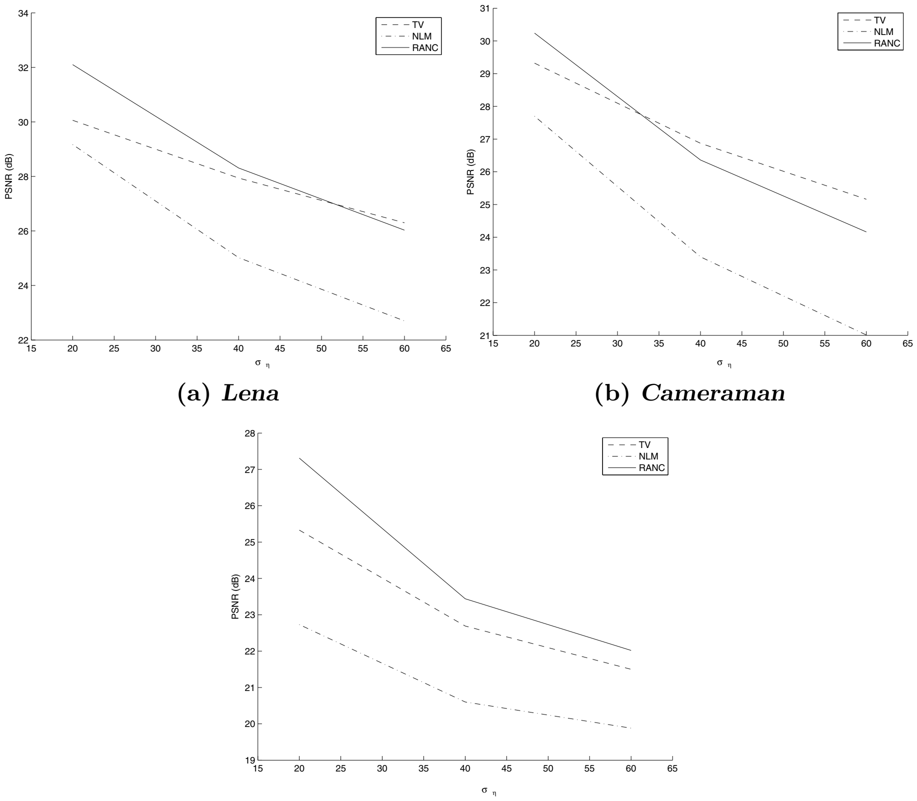
$$w_{r,s}^{(t)} = \frac{d\left(V_r, V_s\right)^{(t)}}{\mu + d\left(V_r, V_s\right)^{(t)}} \tag{15}$$

$$f_r^{(t+1)} = \frac{H(g_r) + \lambda \sum_{s \in W_r} \left(1 - w_{r,s}^{(t)}\right)^2 f_s^{(t)}}{1 + \lambda \sum_{s \in W_r} \left(1 - w_{r,s}^{(t)}\right)^2} \tag{16}$$

The weights $w_{r,s}$ take values between 0 and 1, 0 in case of minimum and 1 when exists a maximum distance between neighborhoods $d(V_r, V_s)$ according to (15), so if μ parameter is equal to zero, the minimizer for (13) is $w_{r,s}$ equal to one and this does not consider the distance between neighborhoods computing a smooth solution in whole image. In resume, the μ parameter controls the piece-wise smoothing restoration.

4 Experimental Results

Experiments were performed with NLM, TV and RANC (using different operators H). Parameters used for NLM are: $h = 40$, search neighborhood W_r size 21×21 and similarity neighborhood V_r size 7×7; for TV $\lambda = 0.2$ were used; for RANC we used $\lambda = 50$, $\mu = 300$, search neighborhood W_r size 11×11 and similarity neighborhood V_r size 3×3. All the parameters were tuned trying to reach the best results in whole cases.



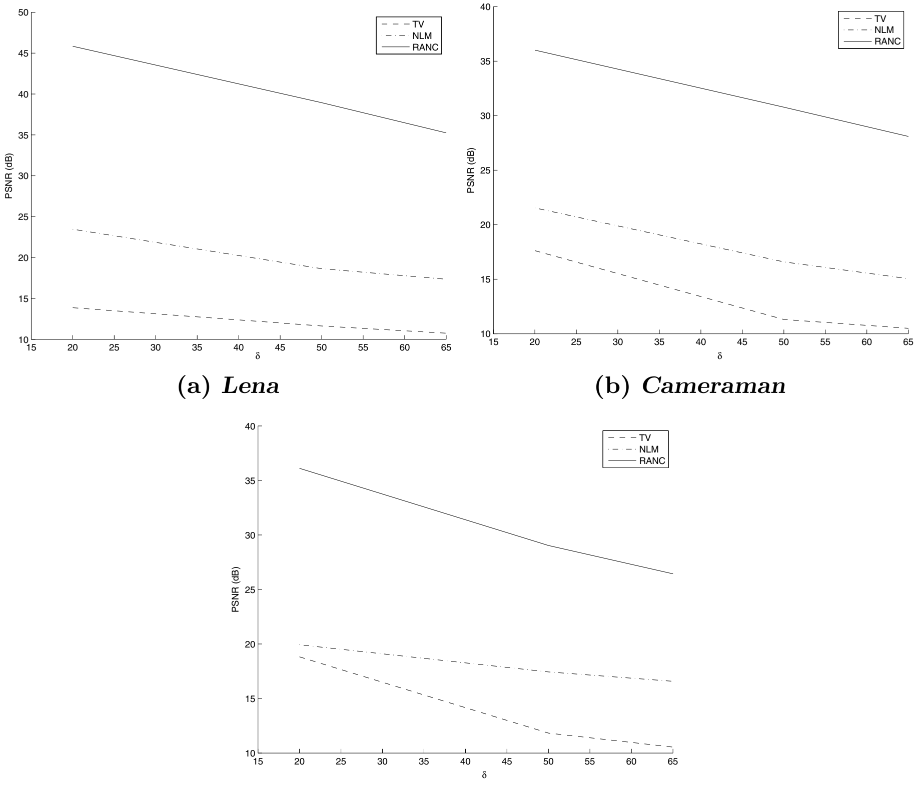
(a) ***Lena***

(b) ***Cameraman***

(c) ***Mandril***

Fig. 4. PSNR comparison for Lena, Cameraman and Baboon images with Additive Gaussian noise

In Fig. 1, we show denoising results of Lena image with zero mean and $\sigma_\eta = 20$ Gaussian noise. We use, in this case, as operator H the identity function, i.e. $H(g) = g$. We can observe that details on the hat are better preserved with RANC, compared with NLM. Regarding to TV restoration, these details are lost, due to the tendency to form flat regions. Fig. 2 shows the obtained restoration for Baboon image, by the three algorithms, the image is altered with salt and pepper noise with 80 % density, and the used operator H is the adaptive median filter proposed in [4], regularization was only performed on pixels detected as noisy. Fig. 3 shows results for Cameraman image with mixed Gaussian ($\sigma_\eta = 20$) and Salt and Pepper (20%) noise, where it can be noticed that performance of RANC is robust. The used operator H is the adaptive median filter. Figures 4(a), 4(c), and 4(b) show the comparative PSNR for the three algorithms, in this case RANC presents better results for low gaussian noise and similar results for images with greater noise deviations. Figures 5(a), 5(c), and 5(b) show the



(a) ***Lena***

(b) ***Cameraman***

(c) ***Mandril***

Fig. 5. PSNR comparison for Lena, Cameraman and Baboon images with Salt and Pepper noise

comparative PSNR results for the Salt and Pepper noise case, in all cases RANC presents best results than NLM and TV. TV does not give good results because restoration is based on Gradient Magnitude and impulsive noise is an amplifier of this magnitude. Figure 6(a) shows a Cardiovascular Ultrasound image and fig. 6(b) the resulting filtered image using RANC with parameters $\lambda = 200$, $\mu = 300$, the final restoration is shaper than the original one but we have to mention that an expert opinion is needed.

The RANC algorithm was tested using more than 30 different images and the results are very similar to images presented in this paper. In all experiments, we use the same parameters which gives an idea of the robustness of RANC algorithm.

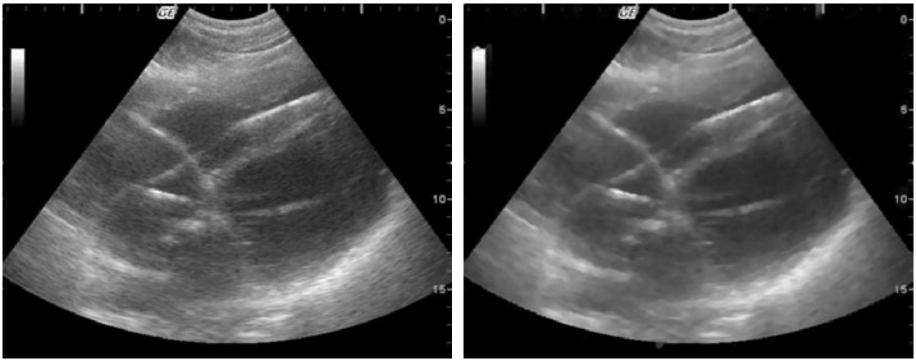
Peak Signal-to-Noise Ratio (PSNR) results between NL-means, TV and RANC for denoising Lena image with mixed Gaussian and Salt and Pepper noises with different standard deviations and densities are shown in Tables 1 and 2 where it can be notice the best performance for RANC in all cases.

Table 1. PSNR for Cameraman image with mixed Additive Gaussian and Salt and Pepper noise

Noise	$\sigma_\eta = 20$ $\delta = 20\%$	$\sigma_\eta = 40$ $\delta = 20\%$	$\sigma_\eta = 20$ $\delta = 40\%$
NLM	21.29	20.61	17.72
TV	18.02	16.29	12.77
RANC	28.12	23.51	26.52

Table 2. PSNR for Baboon image with mixed Additive Gaussian and Salt and Pepper noise

Noise	$\sigma_\eta = 20$ $\delta = 20\%$	$\sigma_\eta = 40$ $\delta = 20\%$	$\sigma_\eta = 20$ $\delta = 40\%$
NLM	19.87	19.76	18.46
TV	18.13	16.61	13.49
RANC	25.77	22.34	23.90



(a) ***Original image*** (b) ***Filtered image with RANC***

Fig. 6. Cardiovascular Ultrasound image restoration

5 Conclusion

We have presented a cost function for image denoising. This function consists of terms that exploit the existence of repetitive structures on images and it considers the influence of highly noisy pixels. Regularization term is controlled by adaptive non-Gaussian weight coefficients. Experimental results have shown that presented proposal has a good performance, compared with some state-of-the-art methods.

References

1. Tomasi, C., Manduchi, R.: Bilateral Filtering for Gray and Color Images. In: ICCV, pp. 839–846 (1998)
2. Buades, A., Coll, B., Morel, J.: A non local algorithm for image denoising. In: Proc. Int. Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 60–65 (2005)
3. Astola, J., Kuosmanen, P.: Fundamentals of Nonlinear Digital Filtering. CRC Press, USA (1997)
4. Hwang, H., Haddad, R.A.: Adaptive median filters: new algorithms and results. IEEE Trans. On Image Processing 4(4), 499–502 (1995)
5. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
6. Rivera, M., Marroquin, J.L.: Adaptive rest potentials: first and second order edge-preserving regularization. Journal of Computer Vision and Image Understanding 88, 76–93 (2002)
7. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Trans. On Image Processing 6(2), 298–311 (1997)
8. Chan, R.H., Ho, C.W., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE Trans. On Image Processing 14(10), 1479–1485 (2005)

Multiple Target Tracking with Motion Priors

Francisco Madrigal, Mariano Rivera, and Jean-Bernard Hayet

Centro de Investigación en Matemáticas (CIMAT)
Guanajuato, Gto., México
{pacomd,mrivera,jbhayet}@cimat.mx

Abstract. This paper presents a particle filter-based approach for multiple target tracking in video streams in single static cameras settings. We aim in particular to manage mid-dense crowds situations, where, although tracking is possible, it is made complicated by the presence of frequent occlusions among targets and with scene clutter. Moreover, the appearance of targets is sometimes very similar, which makes standard trackers often switch their target identity. Our contribution is two-fold: (1) we first propose an estimation scheme for motion priors in the camera field of view, that integrates sparse optical flow data and regularizes the corresponding discrete distribution fields on velocity directions and magnitudes; (2) we use these motion priors in a hybrid motion model for a particle filter tracker. Through several results on video-surveillance datasets, we show the pertinence of this approach.

1 Introduction

Visual target tracking has been the object of intense research in the last years, in a parallel trend to the huge development of visual surveillance (VS). If VS systems have invaded our daily lives, it is remarkable that many of its features still rely on human monitoring of up to dozens video screens. Computer vision techniques, such as motion detection and tracking, have started to be integrated in these systems, but we are still far to have a completely automatic system that would understand how people are moving, give high-level interpretation of the scene, or trigger alerts in case of suspect activities. The present work lies in this context of automatization of video surveillance systems, and in particular in the robust estimation of target trajectories, among mid-dense crowds. In that case, visual tracking is generally possible, but it is made difficult by frequent occlusions between targets and with scene clutter. Multiple, overlapping cameras settings may help to disambiguate among targets identities, but it is expensive, as more hardware is required. Here, instead, we focus on a cheap, *monocular* scheme and aim at tracking as many human targets as possible, in spite of the occlusions they may undergo. To reach that goal, we rely (1) on a probabilistic approach based on an observation model that can cope with partial occlusions, and (2) on a representation of the prior distributions of targets velocities, integrated as a proposal distribution in a particle filtering scheme. This second element, the use of motion prior, is the main contribution of this work, and to our knowledge it had not been proposed before in this form.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 407–418, 2011.

The organization of this paper is as follows: first, in Section 2, we give a short glance on the – huge – existing literature of monocular target tracking; we explain in Section 3 how velocities distribution are estimated, and in Section 4, we propose a particle filter-based scheme that uses these priors as a proposal distribution and motion model. Finally, in Section 5, we present experimental results on standard datasets of video-surveillance and in Section 6, we balance the pros and cons of our approach.

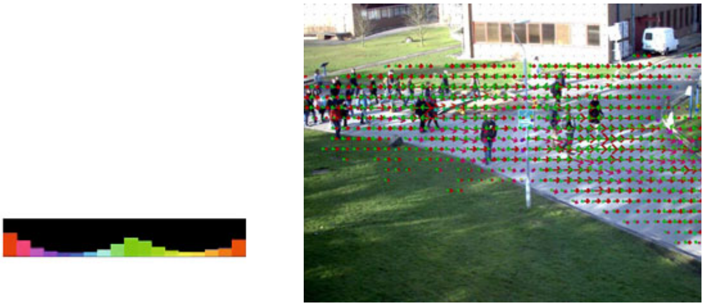
2 Related Work

Target tracking in monocular video streams has generated a great amount of approaches, so that making an overview of the literature is a difficult exercise. Most early works have focused on coping with binary blobs detected from a background modeling algorithm [1], and with using stochastic processes tools imported from the radar community, i.e. by considering objects in motion as undistinguishable "dots". Then standard approaches have made extensive use of object appearance, such as color, geometric moments, spectral properties or, simply, shape.

Seminal appearance-based works include in particular the one of Comaniciu [2], who proposed the concept of Mean Shift tracking, i.e. of tracking the modes of the object likelihood, given its appearance. After the work of [3], in which *particle filters* were introduced for the first time in computer vision, in the context of active contours tracking, Monte-Carlo methods have taken the lead in the literature. Pérez et al. [4], on the one hand, and Nummiaro et al. [5], on the other hand, have proposed the first particle filter trackers based on color histograms, in the context of face tracking. It has proven to be particularly robust, e.g. in video-conferencing applications. The idea, which is still used in this paper, is to use a probabilistic observation model based on the color content of the target to track. Many more extensions have been proposed later on to the original principle of Sequential Monte-Carlo techniques, based on the combination of sampling strategies, probabilistic observation models and probabilistic motion models. On sampling mechanisms (which will be recorded briefly in this paper), novel ideas have been introduced from the statistics community [6], or from the machine learning community [7], among the others. On probabilistic observation models, many works have shown that particle filtering was flexible enough to integrate image-based information of very different nature [8]. On probabilistic motion models, on the contrary, few novel models have been proposed to enhance tracking algorithms, and most systems rely on very simple ones (constant velocity or acceleration, for example). Our work focuses on this part, and studies the use of motion priors learnt from the video monitoring of a scene to improve the motion model.

3 Motion Priors

We build a regularized representation of the distribution of image velocities $\mathbf{v} = (v^m, v^\theta)$ in a monitored scene, where v^m is the velocity magnitude and v^θ its direction, given the position $\mathbf{r} = [x, y]$ in the image, i.e.



(a) (b)

Fig. 1. Velocity map. (a) Velocity orientations histogram for pixel $(560, 220)$. (b) The highest two local maxima in the velocity orientations histogram, for each 20th pixel.

$$p(\mathbf{v}|\mathbf{r}) = p(v^\theta|\mathbf{r})p(v^m|\mathbf{r}, v^\theta). \tag{1}$$

We model $p(v^m|\mathbf{r}, v^\theta)$ as a Gaussian distribution, as, when considering one motion direction at one point, people tend to walk at similar velocity magnitudes. We use a discrete distribution for the more complex $p(v^\theta|\mathbf{r})$. For estimating both distributions, we use motion data samples from video-sequences recorded in the same conditions as encountered during tracking, except that we use a much rougher information than precise, individual target tracking, namely optical flow. The optical flow algorithm to use can be either dense [9] or sparse [10]. In both cases, recollected velocity orientations are integrated into motion orientation histograms, whereas magnitudes are used to estimate the Gaussian conditionals.

From optical flow to velocity orientation histograms. For each frame t of the video sequence, we are given a motion field $\mathbf{v}_t(\mathbf{r}) = (v_t^m(\mathbf{r}), v_t^\theta(\mathbf{r}))$, computed by an optical flow algorithm. Histograms $\mathbf{h}(\mathbf{r})$ of instantaneous velocity directions at an image point $\mathbf{r} = [x, y]$ are incremented at the bin corresponding to the observed orientation $v_t^\theta(\mathbf{r})$, by the quantity

$$v_t^m(\mathbf{r})/(\kappa + v_t^m(\mathbf{r})), \tag{2}$$

where κ serves as a velocity magnitude threshold, in order for static pixels (e.g. from the background) not to be taken into account. Moreover, to limit noisy contributions due to aperture effect, we exclude points for which the minimal value of the eigenvalue of the auto-correlation matrix is inferior to a threshold, as described in [11]. This avoids accumulating incorrect orientations from textureless moving areas.

Regularization of velocity orientation distributions. Let $\mathbf{h}(\mathbf{r})$ be the normalized histogram of instantaneous velocity directions observed in the video-sequence at an image point $\mathbf{r} = [x, y]$, belonging to some image region Ω. Let B be the number of bins in the histogram, and $\mathbf{h}(\mathbf{r})_i$ the value of the i-th bin in this normalized histogram. We define $\log \mathbf{h}(\mathbf{r})$ as the histogram made of the logs of the entries of $\mathbf{h}(\mathbf{r})$, i.e. $[\log \mathbf{h}(\mathbf{r})]_i \overset{\text{def}}{=} \log [\mathbf{h}(\mathbf{r})_i]$.

We estimate a smooth version $\mathbf{d}$ of the log of the histogram field, through the following optimization scheme in the manifold $\mathcal{D} = \{\mathbf{d} \in \mathbb{R}^B \text{ s.t. } \sum_i e^{\mathbf{d}_i} = 1\}$:

$$\min_{\mathbf{d} \in \mathcal{D}} U(\mathbf{d}) = \frac{1}{2} \sum_{\mathbf{r} \in \Omega} \sum_{i,j} \mathbf{W}_{ij} ([\mathbf{d}_i(\mathbf{r}) - \log \mathbf{h}_j(\mathbf{r})]^2 + \lambda \sum_{\mathbf{s} \in N(\mathbf{r})} [\mathbf{d}_i(\mathbf{r}) - \mathbf{d}_j(\mathbf{s})]^2). \quad (3)$$

The first term is the *data* term, fitting the $\mathbf{d}$'s to the collected data. The second one is a smoothness constraint, that makes the histogram at one point $\mathbf{r}$ similar to the ones of its neighbourhood $N(\mathbf{r})$. Terms $\mathbf{W}_{ij}$ encodes the similarity between some different, but close histograms bins, i.e. if the histograms are a bit shifted, they are somewhat similar; moreover, it includes the particular fact of our histograms to be angle histograms, i.e. for which bins are cyclic.

By developing the expression of $U(\mathbf{d})$, and after regrouping terms,

$$U(\mathbf{d}) = C + \frac{1}{2} \sum_{\mathbf{r} \in \Omega} (\mathbf{d}(\mathbf{r})^T \mathbf{W}^{(\mathbf{1})} \log \mathbf{h}(\mathbf{r}) + \sum_{\mathbf{s} \in N(\mathbf{r})} \mathbf{d}(\mathbf{r})^T \mathbf{W}^{(\mathbf{2})} \mathbf{d}(\mathbf{s})), \quad (4)$$

with C a constant, and

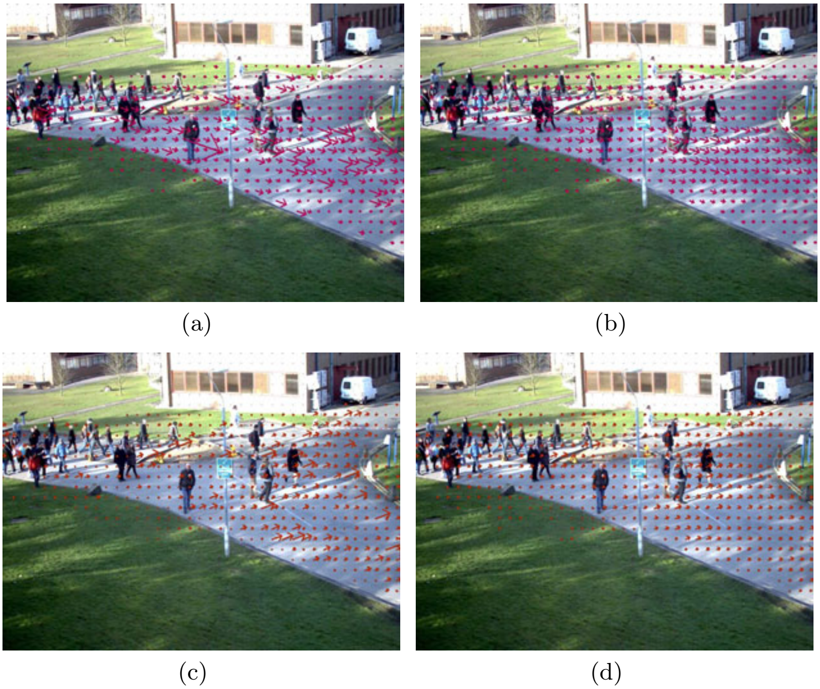
$$\mathbf{W}^{(1)}_{ij} = -2\lambda \mathbf{W}_{ij}, \quad (5)$$

$$\mathbf{W}^{(2)}_{ij} = \begin{cases} -2\lambda \mathbf{W}_{ij} & \text{if } \mathbf{s} \neq \mathbf{p} \\ \begin{cases} (1 + 2V\lambda) \sum_l \mathbf{W}_{il} - 2\lambda \mathbf{W}_{ii} & \text{if } i = j \\ -2\lambda \mathbf{W}_{ij} & \text{otherwise.} \end{cases} & \text{if } \mathbf{s} = \mathbf{p}. \end{cases} \quad (6)$$

Expressed this way, the problem of Eq. 3 can now be solved with classical optimization approaches. Here we adopted a Gauss-Seidel scheme in $\mathbb{R}^B$, in which after each iteration the new estimate for $\mathbf{d}$ is projected on the manifold $\mathcal{D}$ (i.e. re-normalization). In our experiments, some 40 to 60 iterations were necessary to ensure convergence, depending on the chosen value for λ (we set it to 0.5 in ours experiments).

Last, we used the regularized field to fill in image areas where not enough information was available. Typically, this occurs in zones where physical occlusions occur: an example in the datasets we used (see Fig. 2) is the pole at the center of the image. As most people pass behind it, no information is recollected here. However, we can infer the velocity fields in these regions as follows:

1. Form a binary image with data with not enough information and decompose it into convex regions.
2. Get the median axis of each shape and sample pairs of points on the borders, on each side of the median axis.
3. Sample orientations on each of the sampled point pairs.


(a)

(b)

(c)

(d)

Fig. 2. Velocity orientation histogram fields, for two directions i (below and above). (a) and (c) Value in the raw histograms $(\mathbf{v}_k)$. (b) and (d) Regularized histograms after convergence $(e^{\mathbf{d}_i})$. For clarity, data are shown every 20th pixel.

4. Interpolate between these two positions/orientations by a cubic polynomial and generate histogram entries whenever the polynomial curvature stays below a threshold.

With the process above, we can fill in holes in the velocity field, and, as we will show in Section 5, this helps preserving the trajectories continuity.

Results of the histogram calculation and regularization are shown in Fig. 1. We computed the regularized velocity orientation map using $>$ 3000-frames from different sequences of the PETS'2009 dataset [12]. The left image is an example of a histogram for a single pixel, while the right image shows the main two local maxima for the orientation histogram, at each pixel. To speed up the process, we also discarded from the process image regions where no significative motion was observed. In Fig. 1, these are the regions where no arrow is displayed, its complement being Ω. In Fig. 2, we show the regularization results for a particular direction (the one of the arrows), and the lengths of each arrow represent the value of the bin in the histogram. On the left side are the raw histograms, and we can appreciate how, because of the sparseness of the input data in some areas, we may have quite discontinuous orientation distributions around neighbour pixels. For this reason, regularization is critical to provide a smoother, more consistent map.

4 Tracking with Particle Filters and Motion Priors

This section details our particle filter-based tracker and the way it takes benefit from the motion priors computed beforehand.

4.1 Trackers: Definition

As done classically, trackers are defined here as *stochastic filters* that estimate a state relative to the objects of interest, observed through the image. This state includes all the information about the targets needed to predict their posterior state. This could involve their position in the image (i.e. their bounding box center $[x, y]$), their apparent size ($[h, w]$) or their velocity ($[v^m, v^\theta]$). However, as we use particle filtering to perform the inference, it is important to limit the dimensionality of state vectors, to avoid an escalation in the number of required particles. Hence, we opted to set $[h, w]$ as a *deterministic* function of the position $[x, y]$, which is possible when a partial knowledge of the scene is available, as explained for example in [13]. In a few words, to estimate this mapping from $[x, y]$ to target sizes $[h, w]$, we make the assumptions (1) that the objects of interest are pedestrians and (2) that the observed scene is planar. We also suppose that the camera-scene geometry is roughly known, so that an estimate of the projection matrix is available. Hence, the state for a single target is reduced to the target position and its velocity, $\mathbf{X}_t = [x_t, y_t, v_t^m, v_t^\theta]^T$.

To detect new targets and initialize the corresponding trackers, we use a per-pixel statistical background model from a background modeling algorithm [1]. The most likely foreground pixels are grouped into connected components of a binary image (blobs). To manage exclusively pedestrians, we use several heuristics, e.g. a threshold on the size of the connected component (proportionally to the above position-to-scale mapping). We also avoid to validate new targets close to the predicted position of an existing tracker. If the blobs may correspond to a group of people, we use the heuristic of [14], based on the profile of the upper part of the binary region to separate it into individuals. After these filtering steps, we initialize a pedestrian tracker. As for the removal of trackers, e.g. when the corresponding target leaves the camera field of view, we maintain a quality coefficient γ (defined in the next sub-section) for each filter. Whenever γ stays below a threshold for a given period of time, we remove the tracker from the pool of trackers.

4.2 Tracking Pedestrians with Particle Filters

Our trackers are implemented as variants of sequential importance resampling *particle filters* [15]. They require three main elements: a probabilistic observation model, a probabilistic motion model, and a proposal distribution. We explain the three of them in the following.

Observation model. As in [8], the targets appearance model uses color and motion. For each object k to track, we define and update color and motion reference histograms, so that in all frames, we can evaluate the likelihood of a possible target state $\mathbf{X}_{t,k}$ by comparing the histograms $\mathbf{h}_k^f$, computed at $\mathbf{X}_{t,k}$ for feature f, with the reference one, $\mathbf{h}_k^{f*}$. We use seven features in total: color histograms in the H, S, V channels on the upper and lower regions of the target[1] (6 features) and a motion histogram (1 feature). This motion histogram indexes absolute differences between consecutive images.

For each feature f, the comparison with the reference histogram relies on the Bhattacharya distance, referred to as D, between the reference histogram $\mathbf{h}_k^{f*}$ and the current one $\mathbf{h}_k^f(X_{t,k})$. The corresponding likelihood is defined as

$$P(\mathbf{Z}_t|\mathbf{X}_{t,k}) \propto \prod_f \exp\left(-\frac{D^2(\mathbf{h}_k^f(X_{t,k}), \mathbf{h}_k^{f*})}{2(\sigma^f)^2}\right), \tag{7}$$

where σ^f is the expected standard deviation on the Bhattacharya distance for feature f. Setting a large value on the histograms related to the V channel allows for example more robustness to illumination changes. Note that reference histograms are *updated* regularly in order to cope with changes in the image acquisition process (see for example [8] for details on the update mechanism).

Motion model(s). The probabilistic motion model encapsulates the a priori knowledge on how targets move $p(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$. A very simple and common motion model is a constant velocity model

$$\mathbf{X}_{t+1,k} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{I}_{2\times 2} \\ 0 & \mathbf{I}_{2\times 2} \end{bmatrix} \mathbf{X}_{t,k} + S_{x,y}(\bar{\mathbf{X}}_{t,k})\nu_A, \tag{8}$$

where $\nu_A \sim N(\mathbf{0}, \mathbf{\Sigma}_A)$ is a zero-mean Gaussian additive noise with variance $\mathbf{\Sigma}_A$ and $S_{x,y}(\bar{\mathbf{X}}_{t,k})$ is a scale factor deduced from the mapping from $[x, y]$ to target size, described in 4.1. We will refer to this first model as $p^{(1)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$.

A second model is defined with the prior on velocities (Section 3), as

$$p^{(2)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k}) = \gamma p^{(1)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k}) + (1-\gamma)p^{(v)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k}), \tag{9}$$

where $p^{(1)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$ is the motion model in Eq. 8, and $p^{(v)}(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$ a model relying on the estimated velocity prior:

$$\mathbf{X}_{t+1,k} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{I}_{2\times 2} \\ 0 & 0 \end{bmatrix} \mathbf{X}_{t,k} + \begin{bmatrix} 0 \\ \mathbf{v}_{t+1,k}^{(p)} \end{bmatrix}, \tag{10}$$

where $\mathbf{v}_{t+1,k}^{(p)} \sim p(\mathbf{v}_{t+1,k}|\mathbf{r}_{t+1,k})$. The idea is to use this prior as a proposal whenever the filter undergoes difficulties, e.g. because of occlusions, which makes the prediction from the constant velocity model risky, since the state estimation is poor. Hence, the coefficient γ weighting the two distributions is precisely the aforementioned quality measure evaluating the current estimation.

[1] The use of several histograms on different sub-regions is a common method to have some spatial information taken into account.

Proposal distribution. The third fundamental brick of the particle filter is the proposal distribution $q(\mathbf{X}_{t+1,k}|\mathbf{X}_{1:t,k}, \mathbf{Z}_{1:t})$, i.e. the distribution from which samples are generated at each step from the previous one. We chose its “bootstrap” form, where the proposal distribution is the same motion model, i.e. $q(\mathbf{X}_{t+1,k}|\mathbf{X}_{1:t,k}, \mathbf{Z}_{1:t}) = p(\mathbf{X}_{t+1,k}|\mathbf{X}_{t,k})$.

The particle filter in action. Given the previous three ingredients, the particle filter tracker is extremely simple to implement. It maintains a set of weighted particles $\{(\mathbf{X}^{(n)}_{t,k}, \omega^{(n)}_{t,k})_n\}$ and iterate the following steps

1. From the previous set of particles, generate a new set of particles from the proposal distribution, i.e. for all n

$$\mathbf{X}^{(n)}_{t+1,k} \sim q(\mathbf{X}_{t+1,k}|\mathbf{X}^{(n)}_{1:t,k}, \mathbf{Z}_{1:t+1}); \tag{11}$$

   In our case, we will evaluate all the motion models described above as proposal distributions (i.e. $p^{(1)}, p^{(2)}, p^{(3)}$);
2. Update particle weights with the particle state likelihood, the motion model and the proposal, i.e. for all n:

$$\omega^{(n)}_{t+1,k} \propto \frac{p(\mathbf{Z}_{t+1}|\mathbf{X}^{(n)}_{t+1,k})p(\mathbf{X}^{(n)}_{t+1,k}|\mathbf{X}^{(n)}_{t,k})}{q(\mathbf{X}^{(n)}_{t+1,k}|\mathbf{X}^{(n)}_{1:t,k}, \mathbf{Z}_{1:t+1})}\omega^{(n)}_{t,k} = p(\mathbf{Z}_{t+1}|\mathbf{X}^{(n)}_{t+1,k})\omega^{(n)}_{t,k}; \tag{12}$$

3. Compute a quality measure associated to the tracker as the average likelihood among particles.

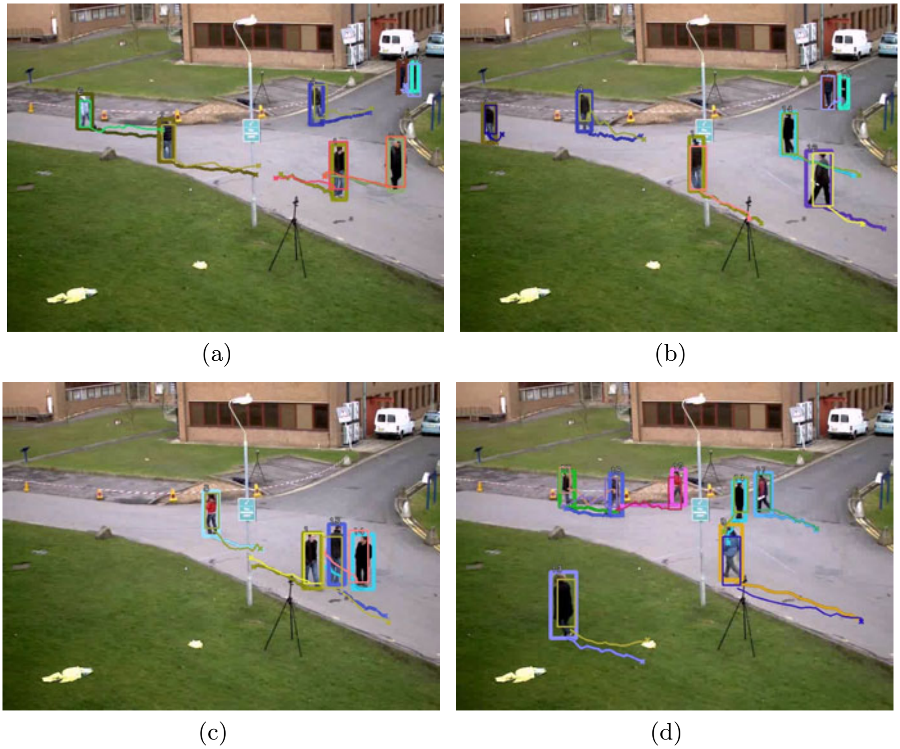
$$\gamma_k = \sum_n p(\mathbf{Z}_{t+1}|\mathbf{X}^{(n)}_{t+1,k})\omega^{(n)}_{t,k}; \tag{13}$$

4. Normalize weights and compute the average state; test the weights variance, approximately $\sum_n [\omega^{(n)}_{t+1,k}]^2$; if it is superior to some threshold, the particle set is degenerate; then proceed to re-sample the particles with replacement.

5 Experimental Results

We present some results of using our approach in real tracking situations. We employed public datasets such as PETS’2009. Since ground truth data is not available, we have manually generated one considering a rectangle around each target in the sequence. The two flavours of the particle filter we propose here (that differ by using or not the motion priors in the probabilistic motion model) are evaluated with a common methodology that has been used to evaluate the trackers performance [12]. Moreover, we also evaluate how the process of filling incomplete prior data improves tracking results.

Illustrations on multiple target tracking with motion prior are presented in Fig. 3. In the represented frames, we draw the feet position of the tracked pedestrian in the 20 previous frames. Thicker lines represent the ground-truth (GT)



(a) (b)

(c) (d)

Fig. 3. A few frames of tracking in sequence S2.L1. The tracks as estimated by our tracker and the ground truth tracks are superimposed.

and thinner lines are our tracking estimation (TE). As it can be noticed, both are very close.

To evaluate our tracking results quantitatively, we used several metrics from the aforementioned evaluation scheme: (1) Normalized Multiple Object Detection Precision (N-MODP), which reflects the target detection rate and precision; (2) Multiple Object Tracking Precision (MOTP), that measures the tracks precision; (3) Sequence Frame Detection Accuracy (SFDA), and (4) Average Tracking Accuracy (ATA), which measures tracks precision but takes more into account the shortening of trajectories. These four indicators take values between 0 and 1 (1 being for the best tracker).

In table 1, we evaluate our tracking system with those metrics. Each row represent the average results obtained in different setups. The first one uses a traditional SIR particle filter tracker with a linear motion model. The second one is a SIR particle filter using the motion prior in the proposal. And the final experiment considers a motion prior with the strategy explained above of filling zones with incomplete information. When comparing the three rows, one can observe that N-MODP and MOTP have similar values but that SFDA and ATA increases substantially when incorporating motion priors. This means that the

Table 1. Results for the S2.L1 sequence. The first row shows the result with different quality indicators for a classic SIR particle filter. The third row shows the results taking into account a motion prior and the last tracking results are obtained with motion priors and the strategy mentioned above of filling zones with incomplete information. Note: All results are the median value of 30 experiments.

Method	SFDA	ATA	N-MODP	MOTP
PF - Linear	0.40	0.42	0.51	0.51
PF - Motion Prior	0.42	0.46	0.51	0.52
PF - Motion Prior Fill	0.45	0.46	0.54	0.54

estimated trajectories, if not more precise, are longer, i.e we can keep the ID of a given pedestrian on longer periods of time. This is what we expected from using priors: in presence of occlusion from other targets or clutter (in which case the estimation of the velocity may be quite noisy) or in situations where the target undergoes non linear motion (i.e. in a zone where pedestrians tend to make sharp turns), the prediction step of the particle filter using motion priors sample particles in a way much closer to what people tend to do at this place than what would generate a too simple or too noisy motion model.

Last, in Fig. 4, we have compared the results of our own tracking strategies with the results obtained by several other authors as reported in [12]. For all

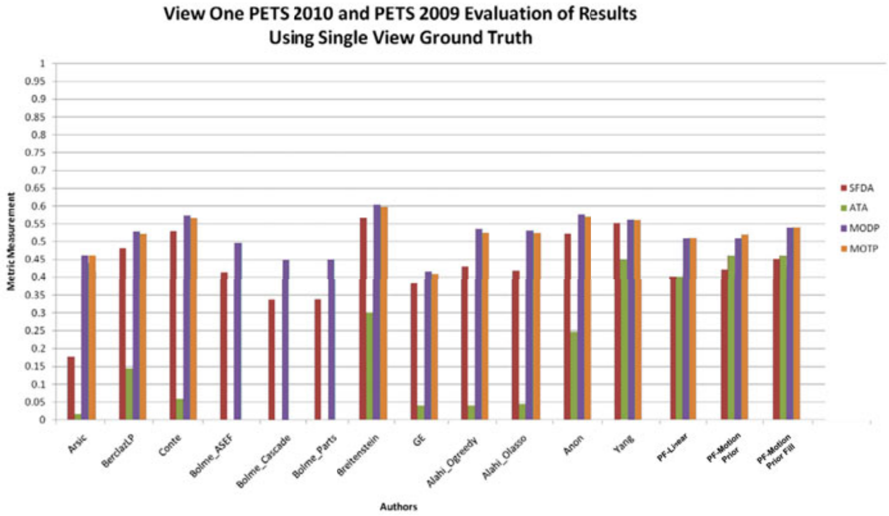


Fig. 4. Performance evaluation of tracking proposed by other authors and our proposal in set S2.L1 (view 1) of PETS 2009 dataset, for the four quality indicators. The last three results use our tracking system with linear motion model, motion prior model and a motion prior model with the strategy of filling zones with incomplete information. Other authors results have been reported in [12].

indicators, the obtained results are quite competitive, and as we can see the most important result is the ATA metric (green column) which reflect the continuity of trajectories, in other words we can follow all targets using less trackers as other approaches.

6 Conclusion

We have described an original enhancement to color and motion-based particle filer tracking that relies on the use of motion priors. We showed how to estimate these prior distributions from optical flow computed on video-sequences grabbed by the same camera from which the camera is done. As in general few data are available for evaluating these distributions, we have proposed a regularization scheme for estimating them and filtering out artifacts. Then we have shown that the use of these priors in the probabilistic motion model of the tracker particle filter allows improvements in the tracker performance, in particular in terms of trajectory targets lengths.

Among our ongoing and future works, we aim at learning higher level information about not only local motion in the scene, but also on long-term information about trajectory, i.e. goals, interactions... based on low-level image information such as optical flow.

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR 1999 (1999)
2. Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: The Proc. of the Seventh IEEE International Conf. on Computer Vision, pp. 1197–1203 (1999)
3. Isard, M., Blake, A.: Condensation-conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
4. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
5. Nummiaro, K., Koller-Meier, E., Svoboda, T., Roth, D., Van Gool, L.: Color-Based Object Tracking in Multi-camera Environments. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 591–599. Springer, Heidelberg (2003)
6. Doucet, A., de Freitas, N., Gordon, N.: Sequnetial Monte Carlo methods in practice. Springer, New York (2001)
7. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
8. Perez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proc. of the IEEE 92(3), 495–513 (2004)
9. Farneback, G.: Fast and accurate motion estimation using orientation tensors and parametric motion models. In: Proc. of Int. Conf. on Pattern Recognition, pp. 135–139 (2000)

10. Bouguet, J.: Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. In: USENIX Technical Conference (1999)
11. Shi, J., Tomasi, C.: Good features to track. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR 1994), pp. 593–600 (1994)
12. Ellis, A., Ferryman, J.: Pets 2010 and pets 2009 evaluation of results using individual ground truth single views. In: Proc. of the IEEE Int. Conf. on Advanced Video and Signal Based Surveillance (AVSS), pp. 135–142 (2010)
13. Madrigal, F., Hayet, J.: Multiple view, multiple target tracking with principal axis-based data association. In: Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance, AVSS (2011)
14. Hu, W., et al.: Principal axis-based correspondence between multiple cameras for people tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence 28, 663–671 (2006)
15. Doucet, A., De Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo methods in practice. Springer, Heidelberg (2001)

Control of a Service Robot Using the Mexican Sign Language

Felix Emilio Luis-Pérez, Felipe Trujillo-Romero, and Wilebaldo Martínez-Velazco

Universidad Tecnológica de la Mixteca,
Division de Estudios de Posgrado
Carretera a Acatlima Km. 2.5 Huajuapan de León Oaxaca
{eluis,ftrujillo,wmartinez}@mixteco.utm.mx

Abstract. This paper presents the results of our research in automatic recognition of the Mexican Sign Language (MSL) alphabet as control element for a service robot. The technique of active contours was used for image segmentation in order to recognize de signs. Once segmented, we proceeded to obtain the signature of the corresponding sign and trained a neural network for its recognition. Every symbol of the MSL was assigned to a task that the robotic system had to perform; we defined eight different tasks. The system was validated using a simulation environment and a real system. For the real case, we used a mobile platform (Powerbot) equipped with a manipulator with 6 degrees of freedom (PowerCube). For simulation of the mobile platforms, RoboWorks was used as the simulation environment. In both, simulated and real platforms, tests were performed with different images to those learned by the system, obtaining in both cases a recognition rate of 95.8%.

Keywords: Robot service, Mexican Sign Language, Neural networks, Pattern recognition.

1 Introduction

Robots are presented more and more as auxiliaries in our daily tasks. Therefore it is imperative that these systems evolve with the purpose of having a better interaction with humans. Several researchers have made advances in this field and have developed a robotic branch named “social robotics” [1]. Social robotics focuses on developing algorithms, methodologies and, of course, robots that meet certain requirements for acceptance by the human being, pursuing human-machine interaction.

One area favored with such developments is the field of entertainment, where several artifacts work as pet robots, and in most cases these are commercial toys. As an example we can mention Robosapien [2], developed by the enterprise WowWee, which offers an approach to children in this sort of technology and more recently with WowWee Alive, artificial pets which react to stimuli from users. Another example of this type of robot is found in Aibo [3], which was developed by Sony and is an electronic dog that has a series of sensors that are used to avoid bumping into objects as well as to sense the human touch.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 419–430, 2011.

Such robots are "programmable" and the user can perform this action to develop different routines. These routines, as it was discussed above, are oriented more towards the recreational field of robotics, but provide a field of experimentation with a more or less intelligent platform and low costs [2, 3, 4, 5].

Moreover, there are developments in universities and laboratories working on the realization of the ideal robot, or rather the interface that allows a better and more natural communication between man and machine. In this area we mention the robot Justin, which is a robot that has two articulated arms and is able to perform some domestic tasks like making coffee. This robot has been developed at the Institute of Robotics and Mechatronics from German Aerospace Centre [6]. Let us cite another robot, Ri-Man [7], which was designed to work in a hospital and provide support for, among other things, transporting patients from one place to another.

Even though the above examples are robots that try to look like human beings, not all of them look totally humanoid. This is the case of Rackham [8], which is a robot developed by the LAAS in France, and served as museum guide in the Cité de l'espace in Toulouse. This robot was able to receive orders via a touch screen and physically guide the user to the place he wanted to visit in the museum. In all moment, Rackham was able to track the user, maintaining visual contact with cameras to avoid losing the user.

COGNIRON [9] is another example, this project was intended to make a companion robot and be curious. In other words, the system would walk into a room and would start to explore it in order to have knowledge about it. This would allow it to know where the objects that could be requested a human user were localized.

As presented above, progress has been significant but still the challenge remains in the manner in which the robots communicate with humans and vice versa. This is due primarily to the way in which the user will tell the robot the action we wants it to perform. So it is required some mechanism to allow the user to give the necessary commands or information to the robot to accurately perform the task.

For this reason we present in this paper the control of a robotic system using the signs of the Mexican Sign Language (LSM) alphabet. The robotic control system consists of a mobile platform equipped with a manipulator with 6 degrees of freedom from Enterprise MobileRobots [10].

The interest in using the MSL is mainly originated by the fact that work is almost non-existent with this language, while most of the developed systems use the coding of the American Sign Language as in [11]. Besides, there is a set of symbols which can serve as standard and be easily usable by anyone familiar with MSL. It must be noted that each country has its own alphabet coding [11, 12, 13, 14, 15] and, even in countries that speak the same language, there are variations in the symbols that represent the same sound.

Like examples of this kind of works we can cited to Munib *et al.*[11] who presented an automatic translation of static gesture of American Sign Language by using Hough transform and neural networks for recognition process. In other way we have to Karami *et al.* [13] in this work they developed a recognition system for Persian Sign Language (PSL) using wavelet transform and neural networks. Al-Jarrah *et al.* [15] has developing a system for automatic translation of gestures of the manual alphabets in the Arabic Sign Language using an adaptive neuro-fuzzy algorithm.

Close to proposed work in this paper, we are going to mentioned to Vargas et al [12] that used a neural network with *backpropagation* learning as recognition system for recognizer the signs alphabet for Colombian Spanish but implemented with an optical approach. In [16] Villa-Angulo and Hidalgo-Silva developed a system that is able to translate the Mexican Sing Language into voice in order to achieve this they using a data-glove for capture the manual expresions with a total classification precision obtained of 97.39%.

2 Method

In order to develop in a satisfactory manner the proposed system, this was developed in different stages: imaging, segmentation, obtaining the descriptor form, learning patterns by the neural network, recognition and validation of the system using a simulation environment and the real robot.

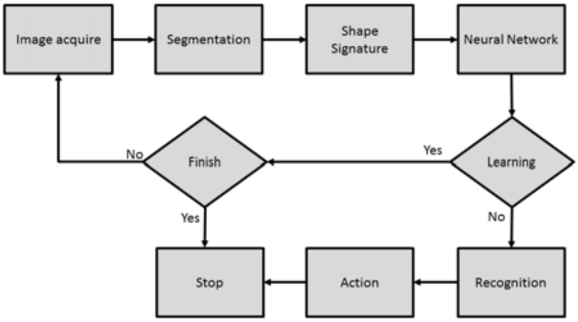


Fig. 1. Recognition system structure

The flowchart of Fig. 1 shows the different stages which are part of the recognition and control system it was implemented. In the paragraphs below we will briefly explain each of the stages mentioned above, giving priority to the main stages of our work.

2.1 Obtaining Images

Only 23 symbols were used (see Fig. 2) of the 27 that comprise the alphabet of the MSL. We choose those whose shape was different from each other. Image sequences were not considered, thus discarding the signs that required movement.

As shown in Fig. 2, the images of the signs were taken using a uniform background. Additionally, the user wore a black garment and only used his right hand. These conditions were established to avoid interference by objects in the wrist (watch, bracelets, etc.), and ensure that the region of interest that is obtained from segmentation belonged only to the hand. It took four different sets of images, each with differences in environment lighting and different background.

An example of the images used to develop the present work is shown in Fig. 3, which shows three distinct images with different photometric characteristics for one of the signs used.

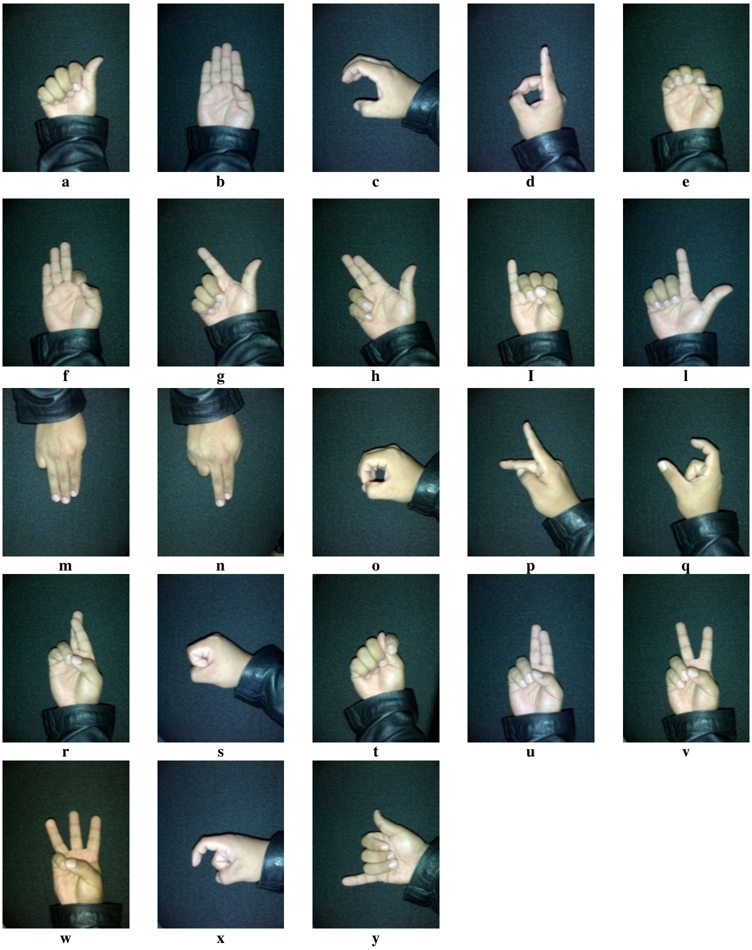


Fig. 2. The 23 symbols of the MSL alphabet

Each of the three images has a brightness variation performed at the time of taking the pictures. Such variation in lighting allowed us to incorporate in some way, a broader spectrum and to discriminate this parameter of enlightenment.

We used four different sets similar to that shown in Fig. 2, each with a different background. These 4 image sets served to create a database of images by varying their parameters using a photometric filter. The filter used was a power function of the form shown in Eq. 1.

$$f(x) = ce^{x} \qquad 0.9 \leq x \leq 1.1 \tag{1}$$

Where c is a constant and x is the parameter to be varied to obtain different intensities in the image that is being processed. The parameter x was varied in a random way in order to obtain different light variation for the input image. In evaluating (1) we have that, if x is less than 1, the image is darker, and if x is greater than 1 is bright (see Fig. 3).

2.2 Segmentation

The first step in the realization of the sign recognition system was the image segmentation by active contours [17]. These contours shape the boundaries between the object, background and other objects in the image. Also allow extraction of the contours of the objects of interest based on models that use a priori information on the shape of objects. These techniques are much more robust to the presence of noise, and allow more complex image segmentation.

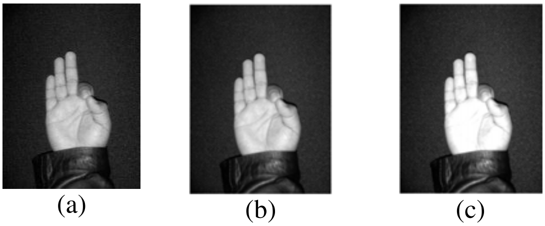
The active contours can be classified as snakes, deformable pattern and dynamic contours. In this paper, snakes were used to perform segmentation. A snake can be defined as an energy-minimizing spline guided by external restraining forces and influenced by image forces, which tends to localize it in features such as lines and edges. It is therefore an active contour that evolves dynamically relevant to the contours of the image.

The snake is represented as a parametric curve by *r(s) = (x(s),y(s))*, with $s \in (0, 1)$. Its functional energy can be expressed as:

$$E^*_{snake}(\boldsymbol{r}) = \int_0^1 E_{int}(r(s))ds + \int_0^1 E_{img}(r(s))ds + \int_0^1 E_{res}(r(s))ds \tag{2}$$

Where E_{int} represents the internal energy of the snake, E_{img} image forces and E_{res} external restraining forces.

There are variations of the original algorithm as presented in [18, 19]. In which the snake is made without edges [18] and [19] is implemented using a vector field. The algorithm used in this work is also a variant which performs segmentation by region. This algorithm is based on the one presented in [20].



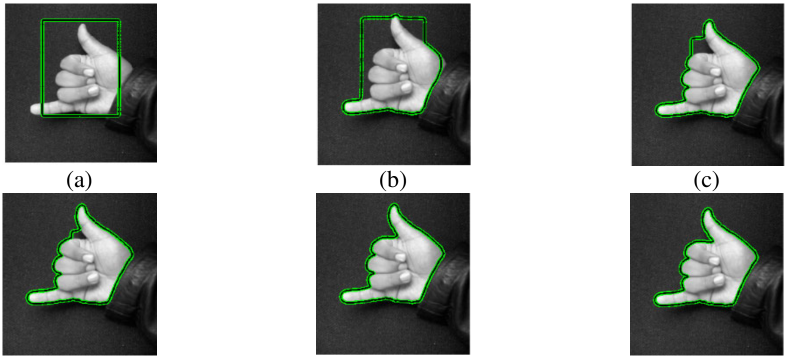
(a) (b) (c)

Fig. 3. Examples for several power settings

We can see in Fig. 4 an example of the different transitions of the snake for the image of MSL symbol representing the letter "y".

2.3 Descriptor

There are many descriptors that use contours as input to generate a corresponding descriptor. Two descriptors of this kind are *shape context* [19] which exploits the contours obtained from a gradient operator, and *shape signature* [20, 21, 22] that use only the external edge of the objects to obtain descriptor.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Different snake transitions (a) initial snake and (f) final contour

Since the snake as result gives us the edge of the object, the *shape signature* was used as descriptor. In our case, the object is the shape of the hand with one sign of the MSL alphabet (Fig. 5).

(a) (b)

Fig. 5. (a) Original image, (b) edges of (a) obtained by snake

In general, obtaining the signature of the object is done by calculating the distances from the object's center of gravity to each one of the points that form the boundary. This yields a histogram of distances as shown in Fig. 6 which is the histogram contour for the symbol shown in Fig. 5a. This histogram consisted of 360 different values which was the input vector for training the neural network.

2.4 Neural Network

We used a neural network composed of three layers with the following configuration: An input layer of 360 neurons, an intermediate layer of 23 neurons, and one neuron in the final layer. The structure of the neural network is shown in Fig. 7.

By choosing this neural network structure, we obtained faster training, especially because of the size of the input vector. In [12] is used an input vector of 18000 values defining an input layer with the same number of neurons. This makes the process slower to analyze portions of the image without information.

In this case, the input vectors are formed by histograms of the shape signature descriptor. Each input vector is associated with an output label ranging from number 1 to number 23. Therefore there is an output of 23 different values, one for each symbol used of the MSL alphabet.

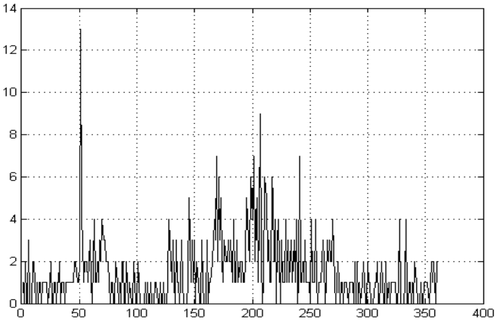


Fig. 6. Distances histogram of the symbol that represents the letter “y”

For the network training, the *backpropagation* algorithm was used. The advantage of *backpropagation* is the fast convergence and robustness over other type of training algorithms [23].

Some of the parameters used for this training were:

- Learning rate of 0.01,
- A maximum of 9000 iterations, and
- A minimum error of 1e-10.

The time it took to the neural network to learn the 1380 pattern vectors and their association with 23 patterns goal was 316 seconds.

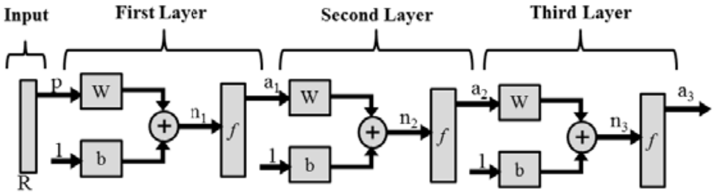


Fig. 7. Neural network structure

3 Results

This section will show the recognition and interpretation results for the set of symbols of the MSL.

We tested the neural network before and after training to observe how the recognition of the symbols of the MSL varied. Fig. 8 shows the graph which was obtained with initial values for the weights of the neural network. In circles are shown the expected values, and in stars those obtained from the output of the network by presenting the symbols of the MSL alphabet. After training the system new graphics were obtained, obtaining the results of Fig. 9. In both cases, we used a series of images corresponding to the full MSL alphabet but varying their photometric characteristics by using the function given in Eq.1.

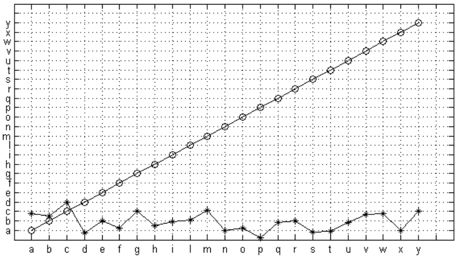


Fig. 8. Neural network output before training

Fig. 9 shows a marked improvement in the recognition of signs for the symbols of the MSL alphabet, in contrast to the output of the network shown in Fig. 8 which does not match any value.

To evaluate the recognition system, a series of different images to those learned by the system were used. By comparing the histograms of learning with the histogram of the symbol to recognize, it was observed some variation while there were many values in the histogram that matched. This makes the neural network to recognize the symbol smoothly as expected. Fig. 10 shows the histogram for the symbol that represents the letter "y" and Fig. 11 shows the difference of the two histograms.

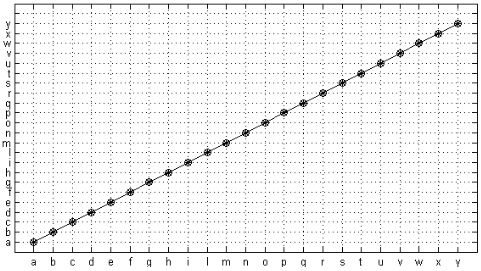


Fig. 9. Neural network output after training

With the acquisition of new images, the recognition phase of the neural network was validated, obtaining the results shown in Fig. 10. In Fig. 12 shows the relationship of the number of times that the system correctly recognized the symbols presented. The experiment was repeated 30 times for each symbol. From the information obtained from these experiments, it was observed that:

- The symbols that were always properly recognize were those that represented the letters a, b, c, d, e, f, g, h, i, m, n, o, r, u, v, x, y.

- The system confused certain symbols, those frequently associated with the same symbol, as in the case of the symbol of the letter l which was classified incorrectly as g.
- The recognition rate was 95.8%.

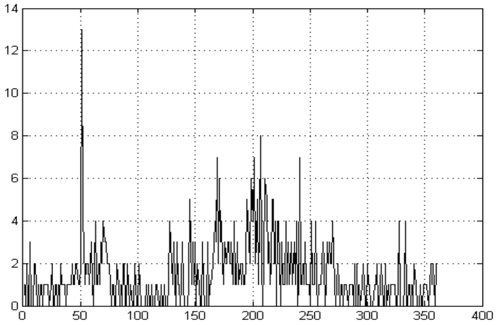


Fig. 10. Distances histogram for the symbol that represents the letter "y" in the recognition phase

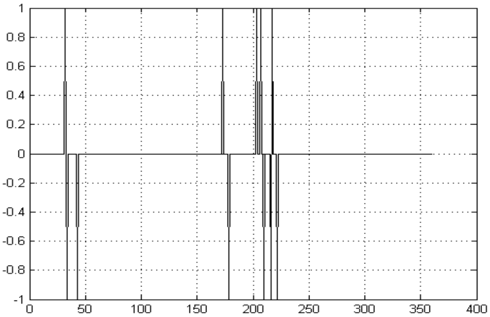


Fig. 11. Error obtained from the difference histogram for the symbol that represents the letter"y"

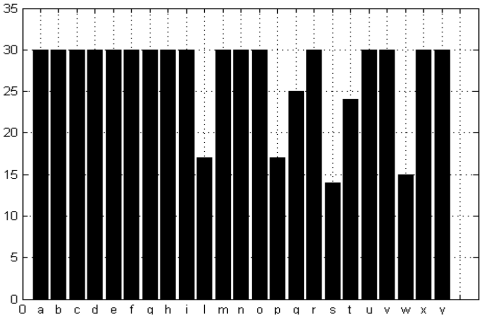


Fig. 12. Bar graph for recognition of the MSL symbols

It must be mentioned that when performing the same experiment with the training images, the system always correctly classified all symbols.

Finally we tested the interpretation of symbols and evaluated it in a robotic context, both in simulation and on a real robot. For the simulation, the simulation environment used was Roboworks [24]. For the real part, the mobile platform Powerbot and the manipulator arm Powercube [10] were used.

A task was assigned to a group of signs of the MSL alphabet. For each of these symbols the actions presented in Table 1 were assigned.

In Fig. 13 the simulated platform performs the routine associated to the symbol “Y”, while the real performance is shown in Fig. 14. Observe that when the system identifies the signal, it sends the interpreted set of subtasks to the platform in order to perform “Y”: serve the drink.

Table 1. Association of symbols with an action to be performed by the robot

Symbol	Action
B	Stop robot
C	Stationary arm configuration
D	Home arm configuration
E	Turning robot
G	Move robot
U	Increase speed
N	Decrease Speed
Y	Glass serving routine

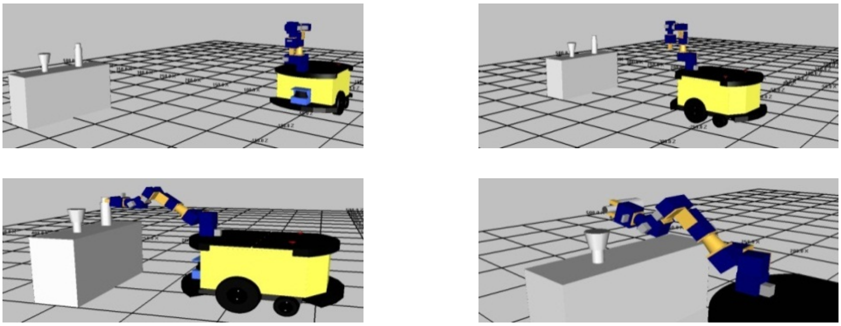


Fig. 13. Simulation of evolution of the task “serve glass”

4 Conclusion and Future Work

The development of this work allowed us to obtain a robotic system able to recognize and interpret a set of signs from the Mexican Sign Language (MSL) alphabet, performing various tasks. While the system was able to recognize 23 symbols of the MSL alphabet, tasks were assigned to only eight of them.

The system achieved 95.8% recognition for learned symbols, due primarily to changes in lighting in the test images and the similarity between some symbols. The developed robot tasks were oriented towards service robotics.

We tested the recognition system using a real robot and a simulation environment, with the same recognition rate. This is because the execution of the robot's movements, whether real or simulated, depends only on the recognition of the signs seen by the system.

Fig. 14. Evolution of the task “serve glass” in a real robot

As future work we expect to assign a task to each symbol of the alphabet in sign language. The assignments may be from a single sign to a sequence of them.

Another consideration is the use of images with variable background, regardless of whether the person that does the marks is wearing a particular garment, as in the images used for this work.

Another field of application of the recognizer could be as translator of Mexican sign language to spoken language.

References

1. Breazeal, C.: Designing Sociable Robots. The MIT Press (2002)
2. Wow Wee Robosapien, `http://www.wowwee.com/en/products/toys/robots/robotics/robosapiens/robosapien`
3. AiboEurope, `http://support.sony-europe.com/aibo/`
4. Samans, J.: The Robosapien Companion: Tips, Tricks, and Hacks (Technology in Action), 1st edn. Springer, USA (2005)
5. Behnke, S., Müller, J., Schreiber, M.: Playing Soccer with RoboSapien. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 36–48. Springer, Heidelberg (2006)
6. Institute of Robotics and Mechatronics fromGerman Aerospace Center, `http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-5471/8991_read-16694/`
7. Robot RI-MAN, `http://rtc.nagoya.riken.jp/RI-MAN/index_us.html`
8. Clodic, A., Fleury, S., Alami, R., Herrb, M., Chatila, R.: Supervision and Interaction: Analysis of an Autonomous Tour-Guide Robot Deployment. In: 12th International Conference on Advanced Robotics, ICAR 2005, Seattle, USA (2005)
9. Cogniron Cognitive Robot Companion, `http://www.cogniron.org/final/Home.php`
10. MobileRobots, `http://www.mobilerobots.com/ResearchRobots/ResearchRobots/PowerBot.aspx`

11. Munib, Q., Habeeb, M., Takruri, B., Al-Malik, H.A.: American sign language (ASL) recognition based on Hough transform and neural networks. Expert Systems with Applications 32(1), 24–37 (2007) ISSN 0957-4174
12. Vargas, L.P., Barba Jiménez, L., Mattos, L.: Sistema de identificación de Lenguaje de Señas usando Redes Neuronales Artificiales. Revista Colombiana de Física 42(2) (2010)
13. Karami, A., Zanj, B., Sarkaleh, A.K.: Persian sign language (PSL) recognition using wavelet transform and neural networks. Expert Systems with Applications 38(3), 2661–2667 (2011) ISSN 0957-4174
14. Paulraj, M.P., Yaacob, S., bin ZanarAzalan, M.S., Palaniappan, R.: A phoneme based sign language recognition system using skin color segmentation. In: 6th International Colloquium on Signal Processing and Its Applications (CSPA), pp. 1–5 (May 2010)
15. Al-Jarrah, O., Halawani, A.: Recognition of gestures in Arabic sign language using neuro-fuzzy systems. Artificial Intelligence 133(1-2), 117–138 (2001) ISSN 0004-3702
16. Villa-Angulo, R., Hidalgo-Silva, H.: A wearable neural interface for real time translation ofspanish deaf sign language to voice and writing. Journal of Applied Research and Technology 3(3), 169–186 (2005) ISSN 1665-6423
17. Blake, A., Isard, M.: Active Contours. Springer, Cambridge (1998)
18. Chan, T., Vese, L.: Active contours without edges. IEEE Transactions on Image Processing 10(2), 266–277 (2001)
19. Lankton, S.: Sparse Field Methods, Technical Report, Georgia Tech (July 2009)
20. Lankton, S., Tannenbaum, A.: Localizing Region Based Active Contours. IEEE Transaction on Image Processing 17(11), 2029–2039 (2008)
21. Belongie, S., Malik, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence (24), 509–521 (2002)
22. Fujimura, K., Sako, Y.: Shape Signature by Deformation. In: Shape Modeling International (SMI 1999). IEEE Computer Society, Aizu (March 1999)
23. Giannarou, S., Stathaki, T.: Shape Signature Matching for Object Identification Invariant to Image Transformations and Occlusion. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds.) CAIP 2007. LNCS, vol. 4673, pp. 710–717. Springer, Heidelberg (2007)
24. Gal, R., Shamir, A., Cohen-Or, D.: Pose-Oblivious Shape Signature. IEEE Transactions on Visualization and Computer Graphics 13(2), 261–271 (2007)
25. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural network design. PWS Pub. (1995)
26. Roboworks, http://www.newtonium.com

Analysis of Human Skin Hyper-Spectral Images by Non-negative Matrix Factorization

July Galeano, Romuald Jolivot, and Franck Marzani

Université de Bourgogne,
LE2I Laboratoire Électronique, Informatique et Image,
UFR Science et Techniques,
BP 47870,21078 Dijon Cedex, France
{july.galeano-zea,romuald.jolivot,franck.marzani}@u-bourgogne.fr

Abstract. This article presents the use of Non-negative Matrix Factorization, a blind source separation algorithm, for the decomposition of human skin absorption spectra in its main pigments: melanin and hemoglobin. The evaluated spectra come from a Hyper-Spectral Image, which is the result of the processing of a Multi-Spectral Image by a neural network-based algorithm. The implemented source separation algorithm is based on a multiplicative coefficient upload. The goal is to represent a given spectrum as the weighted sum of two spectral components. The resulting weighted coefficients are used to quantify melanin and hemoglobin content in the given spectra. Results present a degree of correlation higher than 90% compared to theoretical hemoglobin and melanin spectra. This methodology is validated on 35 melasma lesions from a population of 10 subjects.

Keywords: Blind source separation algorithms, Non-negative Matrix Factorization, human skin absorbance spectrum, Multi/Hyper-Spectral imaging.

1 Introduction

Human skin is a complex multilayered structure composed of several particles that imply different physical phenomena. From an optical point of view, the dominant effects correspond to scattering and absorption. This last effect mostly occurs at melanocyte and erythrocyte cells which are known to contain chromophores. The main light-absorbing pigments present in those cells are melanin and hemoglobin respectively [1]. Melanin is the chromophore of human skin charged mainly for the protection from solar radiation, and in the assessment of skin color. It is also involved in several human skin pathologies such as malignant melanoma, albinism, vitiligo and melasma [2]. Hemoglobin is the pigment related to red blood cells, which are mostly present in vascular densities. This fact has shown the importance of hemoglobin in the study of gastroenterological diseases by improving the classification of colic polypus [3].

Melanin and hemoglobin seem to be the clue for the analysis of different diseases underlying human skin. Nevertheless, since human skin spectrum is

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 431–442, 2011.

the result of the complex light-skin interaction, the obtention of melanin and hemoglobin in a separately way is not an easy task. Due to this, it is important to have a device able to perform two main tasks: to detect human skin absorption/reflectance spectrum, and to discern melanin from hemoglobin.

Several works related to the achievement of those both components can be found. Most of them use spectrometers as the instrument for the acquisition of human skin absorption spectrum. Their difference remains in the way as data is analysed. Among those ways, ones can enumerate Monte Carlo simulations, Kubelka-Munk theory, linear regression, and statistical approaches [4,5,6].

This paper presents a Multi-Spectral approach for the obtainment of human skin reflectance spectrum, and propose the use of a simple Blind Source Separation (BBS) method for the estimation of the amount of melanin and hemoglobin. The human skin data used for the performed analysis was obtained from 10 volunteers presenting with melasma lesions. The later allows us to corroborate the potential use of BSS method in the discernment between melasma and healthy skin zones. The device used for the data acquisition, a Hyper-Spectral system, is presented in section 2. The BSS algorithm, used for the analysis of the obtained Hyper-Spectral data, is presented in section 3. Finally the obtained results are presented and discussed in section 4.

2 Hyper-Spectral Data Adquisition: ASCLEPIOS System

Among the optical instruments used for the study of human skin, it is possible to find color-based instruments (as color cameras) and spectrometers [4]. The first one allows the analysis of a huge area of interest based on the information given by three wide spectral bands, RGB. Although spectrometers allow performing a deeper analysis about skin-light interactions, they allow the analysis in only small areas of interest. The tradeoff is then between size of the area to be evaluated (region of interest ROI) and the accuracy of spectral information for each pixel. This last fact determines the kind of analysis which can be performed.

Multi-Spectral and Hyper-Spectral Imaging (MSI/HSI) systems are presented then as a way to overcome the problematic size of ROI-kind of analysis. Since it acquires images at different spectral bands, shorter than the conventional RGB systems, MSI-HSI systems allow then a spectral analysis in a big area of interest of the object under study. This information gives not only a merely information and improvement about skin color, but also information about how skin's components interact with light.

For the purpose of the present work, we use a system called ASCLEPIOS (Analysis of Skin Characteristics by Light Emission and Processing of Images Of Spectrum), which is an innovative system since it evolves from a MSI into a HSI system without the need for an increased number of wavelengths [7]. As depicted in figure 1, the device consists of a camera, and a rotating wheel with a set of 10 filters. Those filters are between the wavelengths of 430 to 780 nm, each one with Full Width at Half Maximun (FWHM) of 80 nm. In this way the object of interest is illuminated at 10 different spectral bands leading to a MSI. Then, by using a software based on artificial neural networks [8], an Hyper-Spectral cube of the object under study is obtained. The cube is then the HSI. The x,y coordinates of the HSI correspond to the spatial dimension giving the reflectance value at each pixel coming from the camera. The z coordinate is the spectral dimension corresponding to 36 different spectral bands with a FWHM of 10 nm. In this way, a reflectance spectrum composed of 36 values between 430 and 780 nm, is obtained at each pixel of the HSI.

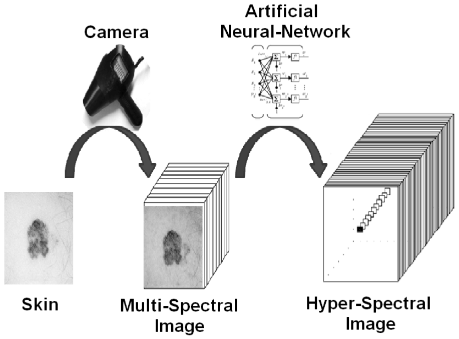


Fig. 1. Scheme of the used system called ASCLEPIOS

This Hyper-Spectral device has been validated on a population composed of 150 healthy participants from five different Skin Photo Types (SPT) at different body locations. The validation was performed by comparing data acquired from a commercial spectrophotometer with the spectra processed from averaging the obtained HSI. The results revealed that the system is able to provide HSI with a Goodness of Fit Coefficient (GFC) superior to 0,997 for the average of all SPT for each location. This means that ASCLEPIOS system provides accurate Hyper-Spectral Images which can be effectively used for analysis of skin reflectance spectra [9].

3 Hyper-Spectral Data Modeling and Analysis

The acquired Hyper-Spectral data are analysed by a linear BSS method. Such method represents the given data as a weighted sum of source components. The later implies linearity in the phenomena to be studied. Since our goal is to obtain the principal components of human skin from a measured reflectance spectrum, it is important to represent the physical phenomena of light-skin interaction in a linear way. For that representation, the study presented in this article is based on the modified Beer-Lambert law.

3.1 Physical Approach: Light-Skin Interaction

Human skin is presented as a scattering multi-layered media composed of different pigments. As depicted in figure 2, the principal pigments of human skin, melanin and hemoglobin, are present in epidermis and dermis respectively [10,11].

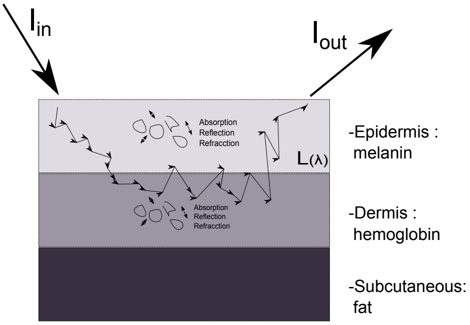


Fig. 2. Light-skin interaction: when light interacts with the different layers present in skin, absorption, reflection and refraction occurs at the pigments. The principal pigments present in human skin are considered to be melanin and hemoglobin.

When ligth interacts with skin, ligth travels through the different layers where scattering, absorption, and reflection occurs at the pigments. As a result, light travels inside human skin through a geometrical path dependant of the wavelentgh ($L(\lambda)$). Since ASCLEPIOS sytem detects reflectance $R(\lambda)$ spectrum as the ratio of incident to reflected energy, absorbance spectrum $A(\lambda)$ can be deducted from reflectance by equation 1 [12]:

$$A(\lambda) = -10\log(R(\lambda)). \tag{1}$$

In optical absorption terms, the modified Beer-Lambert law holds. The total absortion of human skin can be determined as the contribution of the absorption values present at the different layers. This is expressed by equation 2 [11].

$$A(\lambda) = \sum_{i=1}^{n} \varDelta A_i(\lambda) = \sum_{i=1}^{n} C_i \epsilon_i(\lambda) L(\lambda) + G. \tag{2}$$

According to the relation 2, the change in absorbance at layer i ($\varDelta A_i(\lambda)$) is related to the molar absorption coefficient of pigment i ($\epsilon_i(\lambda)$), the concentration C_i, and the mean path length $L(\lambda)$. G are the losses due to the components not considered in the model.

In this work, BSS allow then to obtain the spectral absorption of dermis and epidermis. Those spectra are considered then as an approximation to the molar absortion coefficient of melanin and hemoglobin respectively, which average spectra are presented in figure 3.

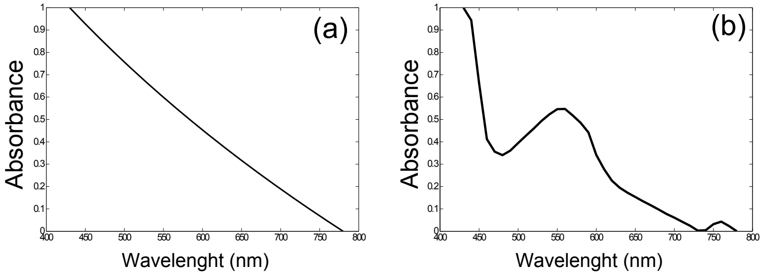


Fig. 3. Average absortion spectrum of: (a) melanin, and (b) hemoglobin

3.2 Blind Source Separation Applied to Dermatology

The goal of BSS algorithms is to decompose a given signal in its main sources. Some of their uses are for example the analysis of biomedical signals, telecommunications, and Multi-Hyper-Spectral Imaging. The points to be considered before using linear BSS algorithms correspond to a previous knowledge of the expected results, and to ensure linearity in the physical phenomena to be represented [13].

Since human skin absorption spectra can be represented as a linear combination of components, linear BSS techniques could be useful in separating them. For such purpose, different methods can be found such as Independent Component Analysis (ICA) and Non-negative Matrix Factorization (NMF). Their implementation depends on the kind of materials used for the data acquisition. If color based instruments are used, a spatial approach is more suitable. In the case of Hyper-Spectral systems a spectral approach is more convenient. In spatial approach, images obtained at different spectral bands can be seen as a linear combination of source images. As a example ones can mention the work of Tsumura et al, who used ICA methods to separate the spatial distributions of melanin and hemoglobin in skin from a color image [6].

In the case of Hyper-Spectral systems, more than 3 values are obtained for each pixel. As depicted in figure 4, the spectrum at each pixel of the Hyper-Spectral cube can be seen as the linear combination of spectral sources, in this case melanin and hemoglobin.

Although ICA can be useful in skin component decomposition from a spectral point of view [14,15], here we center the discussion on the use of simple BSS methods such as NMF. The latter have been widely used for the study of geological components [13,16]. Nevertheless, to our knowledge, NMF are not so much used in the study of human skin components. Due to this lack, here we evaluate the potential use of NMF in the decomposition of HSI.

The implementation of NMF algorithms is supported on the non-negativity of the data to be evaluated. In our case, this constrain is related to the physical meaning of the HSI obtained with ASCLEPIOS system.

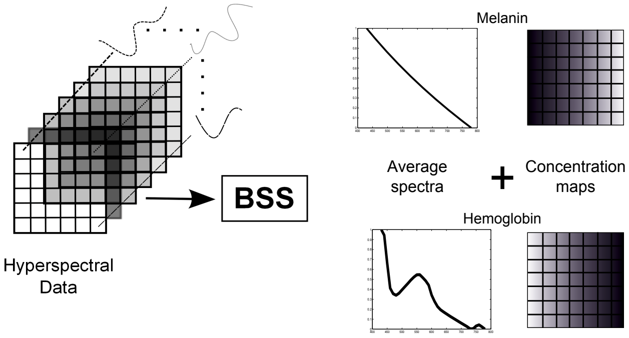


Fig. 4. Spectral approximation in BSS algorithms. The spectrum obtained at each pixel of a Hyper-Spectral cube is observed as the weighted sum of principal components. In our case, these components are considered to be melanin and hemoglobin. BSS algorithms obtain the average spectra of those principal components, together with their respective quantification at each pixel of the Hyper-Spectral cube (concentration maps).

From a mathematical point of view, the idea in NMF is to approximate a given $n \times m$ matrix Y, with $Y_{ij} \geq 0$, to the product of two non-negative matrices $W \in R^{n \times r}$ and $H \in R^{r \times m}$ $(Y \approx WH)$ [17,18,19].

The typical way to find those non-negative matrices W and H is minimizing the difference between Y and WH by:

$$f(W,H) \equiv \frac{1}{2} \|Y - WH\|_F^2 \tag{3}$$

where $\|\cdot\|$ is the Forbenious norm.

As it is well known in the domain, a multiplicative update rule has been proposed by Lee and Seung [17] to solve the difference denoted by equation 3. This multiplicative update is giving by:

$$\begin{aligned} H_{a,u} &\leftarrow H_{a,u} \frac{(W^TY)_{a,u}}{(W^TWH)_{a,u}} \\ W_{i,a} &\leftarrow W_{i,a} \frac{(YH^T)_{i,a}}{(WHH^T)_{i,a}}. \end{aligned} \tag{4}$$

The function denoted by 3 can be modified in several ways according to the application. So that, penalties can be added in order to enforce sparseness or smoothness in the obtained matrices W and/or H [16,20]. In our case we used smoothness pelnaty in matrix H. In this way the multiplicative update presented in relation 4 becomes:

$$\begin{aligned} H_{a,u} &\leftarrow H_{a,u} \frac{(W^TY)_{a,u} - H_{a,u}}{(W^TWH)_{a,u}} \\ W_{i,a} &\leftarrow W_{i,a} \frac{(YH^T)_{i,a}}{(WHH^T)_{i,a}}. \end{aligned} \tag{5}$$

On the scope of this work, the $n \times m$ matrix Y is the bidimensional representation of the Hyper-Spectral cube obtained with ASCLEPIOS system. The number of columns of matrix Y corresponds to the number of spectral bands, 36 in this case. Each column of this matrix represents the spatial distribution of absorption values at the given spectral band.

Each line of matrix H contains the calculated absorption spectra of melanin and hemoglobin repectively. The theoretical spectra are considered to be the ones presented at figure 3. Finally, matrix W presents in each column the estimated quantification of melanin and hemoglobin at each pixel of the ROI.

In the following, matrix Y is considered as the measured Hyper-Spectral cube, and the multiplication $W \times H$ as the estimated one.

4 Results and Analysis

NMF algorithm is used for the study of different skin spectra. Those spectra are considered as the linear combination of melanin and hemoglobin components. The analysis is done in melasma lesions and healthy skin areas from 10 patients. The results are evaluated in three different ways: comparing the measured and estimated Hyper-Spectral cubes, comparing the theoretical and calculated absorption spectra of melanin and hemoglobin, and analysing the quantification of melanin with respect hemoglobin.

In this paper we present in figures 5 and 6, the results obtained for 2 patients. These two patients are considered to be representative of the total evaluated population. For each one of these figures, two sets of four pictures are presented. Set (a) corresponds to the results obtained for the healthy ROI, and set (b) are the results for the melasma lesion. For each set, the upper left image denotes by a white square the ROI of the skin's area under evaluation. The upper right image presents by continuous and dashed line the measured and estimated absorbance spectra in one pixel of the ROI. In most of cases it is possible to observe the fine congruence between both curves. The calculated absorption spectra of melanin and hemoglobin are presented in left down image. In the case of melanin, it is possible to observe how the obtained curve presents a decay of almost 50% in absorbance around 550 nm, which is in coherence with the theoretical result. For hemoglobin, results present the characteristic absorption peaks around 450 and 570 nm, as presented in theoretical spectrum. Finally, the right down image presents in a histogram the normalized concentration of melanin and hemoglobin in the evaluated area. Since the model presents skin absorption spectrum as a linear combination of melanin and hemoglobin, components' concentration values are interpreted to be relative. In this way, ROI corresponding to melasma lesions present for melanin a histogram with a peak at a concentration value higher than the histogram's peak of hemoglobin. This fact suggests us that for the presented method, melasma lesions present higher concentration of melanin with respect hemoglobin (figures 5b and 6b). In the case of ROI corresponding to healthy skin areas, the histogram's peak of hemoglobin is at a higher (6a) or same (figure 5a) concentration values than the ones from melanin. The results are coherent with the histological cause of melasma: an increased amount of melanin component [21].

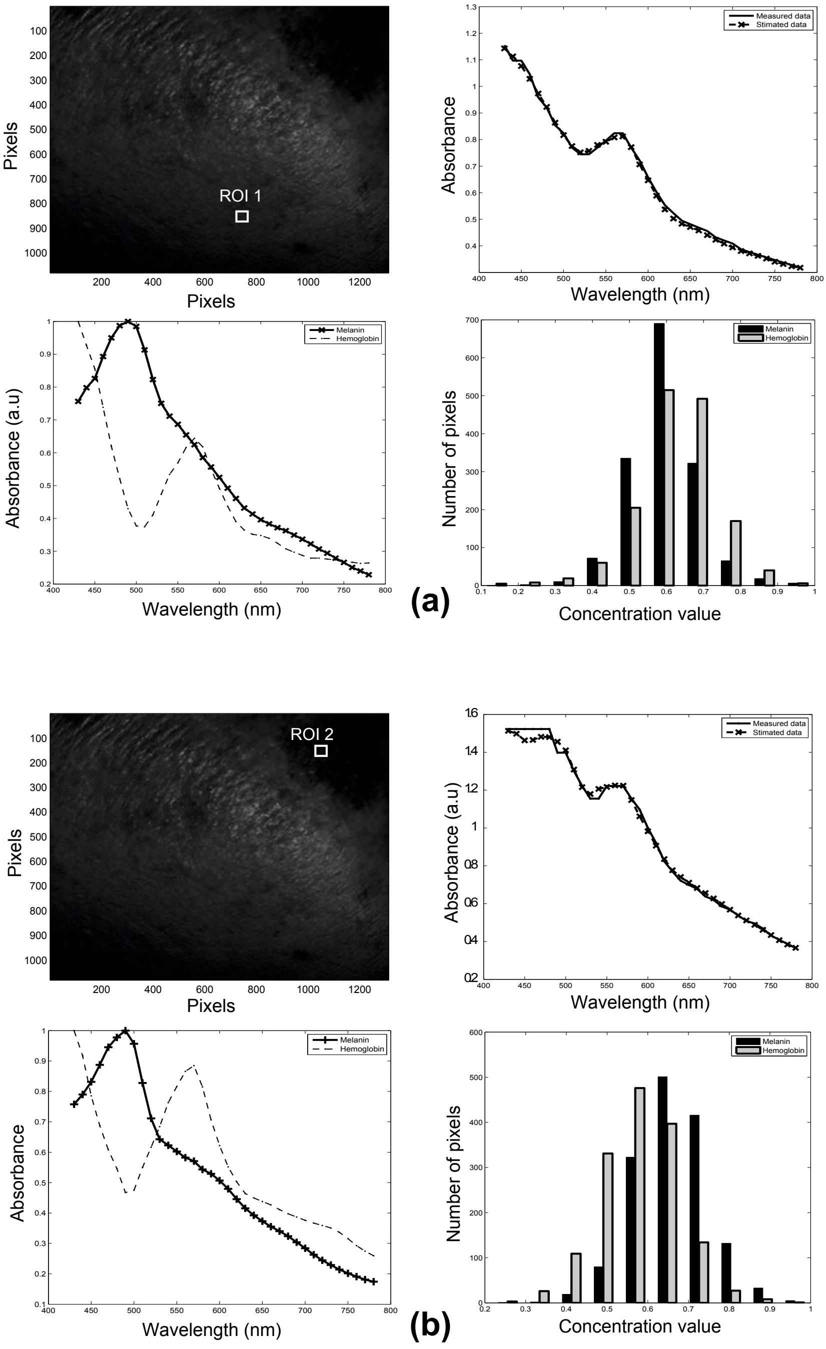


Fig. 5. Results in (a).Healthy skin area, and (b) Melasma lesion. (a-b)*i* Region of interest ROI. (a-b)*ii* Measured and estimated absorption spectra in one pixel of the Hyper-Spectral cube. (a-b)*iii* Estimated melanin and hemoglobin. (a-b)*iv* Histogram of melanin and hemoglobin concentrations.

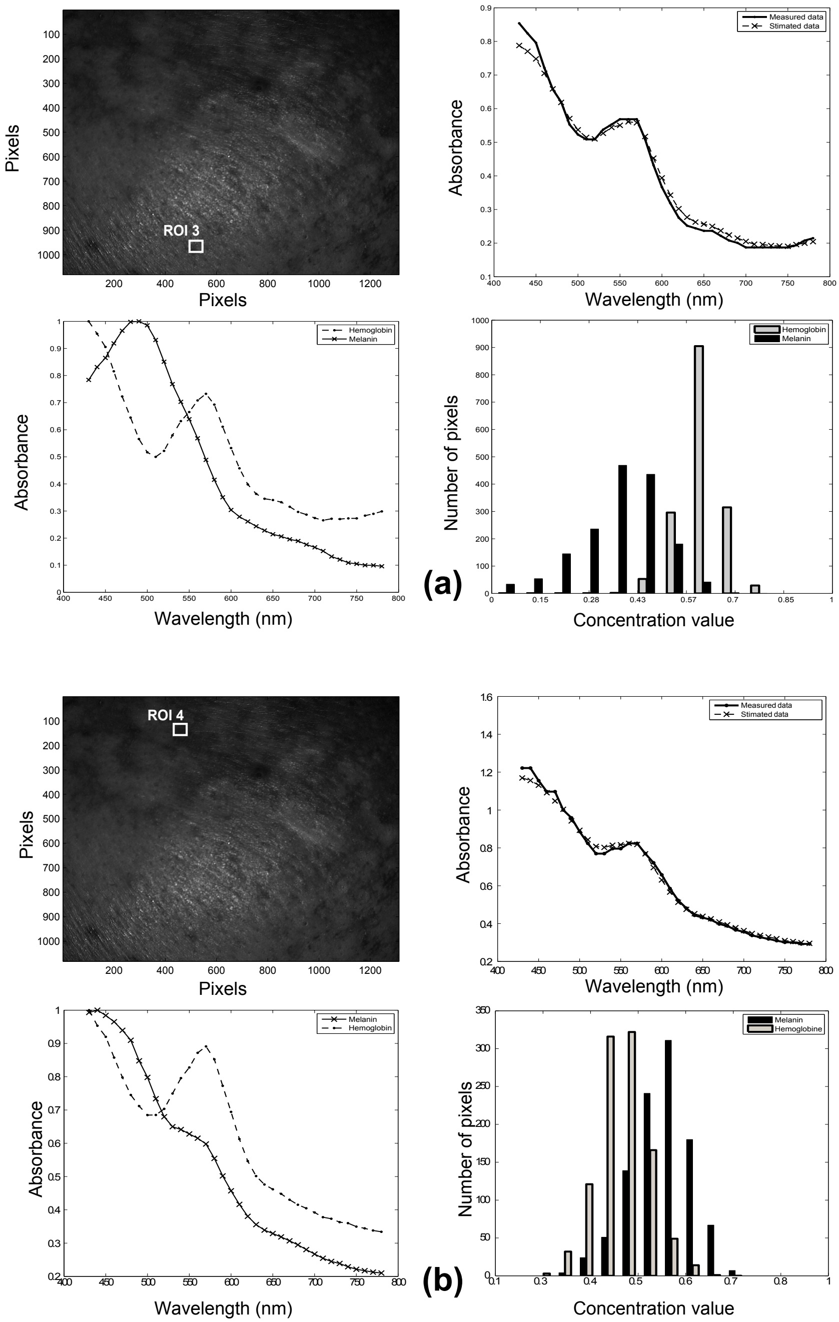


Fig. 6. Results in (a).Healthy skin area, and (b) Melasma lesion. (a-b)*i* Region of interest ROI. (a-b)*ii* Measured and estimated absorption spectra in one pixel of the Hyper-Spectral cube. (a-b)*iii* Estimated melanin and hemoglobin. (a-b)*iv* Histogram of melanin and hemoglobin concentrations.

In a numerical way, three coefficients of correlation are calculated for each analysed area: a first one corresponds to the degree of correlation between the calculated and the measured Hyper-Spectral cubes; second and third one are calculated between the theoretical and estimated melanin-hemoglobin absorbance spectra. Correlation was evaluated using equation 6 [22].

$$r = \frac{\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{N}((X_i - \overline{X})^2)}\sqrt{\sum_{i=1}^{N}((Y_i - \overline{Y})^2)} \tag{6}$$

where X is the estimated data and Y is the theoretical one. $\overline{X}$ and $\overline{Y}$ are the mean value of the estimated and theorical data repectively. The results, which are given in table 1, present in most cases a degree of correlation higher than 0.9.

Table 1. Correlation Coefficients

	Correlation Result		
Sample Number	Hyper-Spectral	Melanin	Hemoglobin
ROI 1	0.99	0.93	0.91
ROI 2	0.99	0.95	0.94
ROI 3	0.99	0.94	0.94
ROI 4	0.99	0.98	0.94

5 Conclusions and Further Work

Non-negative Matrix Factorization algorithm has been applied in the study of human skin absorbance spectrum from 10 patients presenting melasma lesions. The data correspond to a Hyper-Spectral cube obtained with ASCLEPIOS system. The use of a multiplicative update approach demonstrated its capacity in estimating in a ROI, two of the principal pigments present in skin: melanin and hemoglobin. Also, the relative quantity or concentration of these two pigments is estimated with the mentioned algorithm. The estimated pigments together with their relative concentrations lead to a stimated Hyper-Spectral cube. In most cases, this cube presents a degree of correlation close to 90% with respect to the one obtained from ASCLEPIOS system. In the same way, a degree of correlation higher than 90% is obtained between the estimated and theoretical absorption spectra of melanin and hemoglobin. It has been also shown that melasma lesions present higher concentrations of melanin with respect to hemoglobin. In the case of healthy skin areas, hemoglobin concentration is higher or equal to melanin one. Results agree with the histological cause of melasma: melasma is an hyperpigmentation caused by an increment in melanin [21]. The use of such spectral decomposition of the data obtained from Hyper-Spectral system seems to be a useful tool for the study of human skin illnesses.

Nevertheless, the presented work needs further analysis for the study of the two major components underlying melanin and hemoglobin pigments: eumelanin-pheomelanin, and oxy-deoxy hemoglobin respectively. Human skin phantoms can be also evaluated with the aim to corroborate the effectiveness of NMF algorithms.

Acknowledgments. We would like to thank the cooperation given by Dr. Roshidah baba and Dr. Noorlaily from the Department of Dermatology at Hospital Kuala Lumpur. Also we would like to acknowledge the assistance given by Professor Ahmad Fadzil and Hermawan Nugroho from Universiti Teknologi Petronas; and Dr. Norashikin from the faculty of Medicine and Health Sciences at Universiti Putra Malaysia.

The authors would like also to thank the financial support provided by *Conseil Regional de Bourgogne*-France, and *Fond Européen de Développement Regional (FEDER)*-France.

References

1. Igarashi, T., Nishino, K., Nayar, S.K.: The appearance of human skin. Columbia Univ., New York, Tech. Rep. CUCS-02405 (2005)
2. Zonios, G., Dimou, A., Bassukas, I., Galaris, D., Tsolakidis, A., Kaxiras, E.: Melanin absorption spectroscopy: new method for noninvasive skin investigation and melanoma detection. Journal of Biomedical Optics 13, 14017 (2008)
3. Stehle, T., Auer, R., Gross, S., Behrens, A., Karssemeijer, N., Giger, M.L.: Classification of colon polyps in nbi endoscopy using vascularization features. In: Proc SPIE, vol. 7260, pp. 2s1–2s12 (2009)
4. Tuchin, V.V.: Tissue optics: light scattering methods and instruments for medical diagnosis. In: SPIE-International Society for Optical Engineering (2007)
5. Shimada, M., Masuda, Y., Yamada, Y., Itoh, M., Takahashi, M., Yatagai, T.: Explanation of human skin color by multiple linear regression analysis based on the modified Lambert-Beer law. Optical Review 7(4), 348–352 (2000)
6. Tsumura, N., Haneishi, H., Miyake, Y.: Independent-component analysis of skin color image. JOSA A 16(9), 2169–2176 (1999)
7. Jolivot, R., Vabres, P., Marzani, F.: Reconstruction of hyperspectral cutaneous data from an artificial neural network-based multispectral imaging system. Computerized Medical Imaging and Graphics 35(2), 85–88 (2011)
8. Mansouri, A., Marzani, F.S., Gouton, P.: Neural networks in two cascade algorithms for spectral reflectance reconstruction. In: IEEE International Conference on Image Processing, ICIP 2005, vol. 2, pp. 718–721 (2005)
9. Jolivot, R., Vabres, P., Marzani, F.: Validation of a 2d multispectral camera: application to demartology/cosmetology on a population covering five skin phototypes. In: SPIE 2011 (2010)
10. Martelli, F., Del Bianco, S., Ismaelli, A., Zaccanti, G.: Light Propagation through Biological Tissue and Other Diffusive Media (2010)
11. Shimada, M., Yamada, Y., Itoh, M., Yatagai, T.: Melanin and blood concentration in human skin studied by multiple regression analysis: experiments. Physics in Medicine and Biology 46, 2385 (2001)
12. Anderson, R.R., Hu, J., Parrish, J.A.: Optical radiation transfer in the human skin and applications in in vivo remittance spectroscopy. In: Bioengineering and the Skin: Based on the Proceedings of the European Society for Dermatological Research Symposium, p. 253. Springer, Heidelberg (1981)
13. Keshava, N., Mustard, J.F.: Spectral unmixing. Signal Processing Magazine, IEEE 19(1), 44–57 (2002)

14. Mitra, J., Jolivot, R., Vabres, P., Marzani, F.S.: Source separation on hyperspectral cube applied to dermatology. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7624, p. 116 (2010)
15. Mitra, J., Jolivot, R., Vabres, P., Marzani, F.: Blind source separation of skin chromophores on a hyperspectral cube. In: ISBS International Symposium (2009)
16. Pauca, V.P., Piper, J., Plemmons, R.J.: Nonnegative matrix factorization for spectral data analysis. Linear Algebra and its Applications 416(1), 29–47 (2006)
17. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
18. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Computation 19(10), 2756–2779 (2007)
19. Wang, W., Zou, X.: Non-negative matrix factorization based on projected nonlinear conjugate gradient algorithm. In: ICA Research Network International Workshop (ICARN 2008), pp. 5–8 (2008)
20. Comon, P., Jutten, C.: Handbook of Blind Source Separation: Independent component analysis and applications. Academic Press (2010)
21. Miyachi, Y.: Therapy of Skin Diseases. Springer, Heidelberg (2009)
22. Stone, J.V.: Independent component analysis: a tutorial introduction. The MIT Press (2004)

Similarity Metric Behavior for Image Retrieval Modeling in the Context of Spline Radial Basis Function

Leticia Flores-Pulido[1], Oleg Starostenko[2], Gustavo Rodríguez-Gómez[3],
Alberto Portilla-Flores[1], Marva Angelica Mora-Lumbreras[1],
Francisco Javier Albores-Velasco[1], Marlon Luna Sánchez[1],
and Patrick Hernández Cuamatzi[1]

[1] Universidad Autónoma de Tlaxcala , Tlaxcala, C.P. 90180, México
[2] Universidad de las Américas Puebla, Puebla, C.P. 72820, México
[3] Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, C.P. 72840, México
{leticia.florespo,oleg.starostenko}@udlap.mx, grodrig@inaoep.mx,
{alberto.portilla,angelicaml,javier.albores}@gmail.com,
marlon@ingenieria.uatx.mx, trulypatrick@hotmail.com

Abstract. In this paper, the analysis of similarity metrics used for performance evaluation of image retrieval frameworks is provided. Image retrieval based on similarity metrics obtains remarkable results in comparison with robust discrimination methods. Thus, the similarity metrics are used in matching process between visual query from user and descriptors of images in preprocessed collection. In contrast, the discrimination methods usually compare feature vectors computing distances between visual query and images in collections. In this research, a behavior of spline radial basis function used as metric for image similarity measurement is proposed and evaluated, comparing it with discrimination methods, particularly with general principal component analysis algorithm (GPCA). Spline radial basis function has been tested in image retrieval using a standard image collections, such as COIL-100. The obtained results using spline radial basis function report 88% of correct image retrieval avoiding a classification phase required in other well-known methods. The discussion of tests with designed Image Data Segmentation with Spline (IDSS) framework illustrates optimization and improvement of image retrieval process.

Keywords: Splines, Radial Basis Functions, Image Retrieval Modeling, Data Sets.

1 Introduction

The performance of image retrieval is evaluated by applying similarity metrics, which provide quantitative analysis of matching input user visual queries with images in standard collections. There are several metrics adapted in VIR for computing similarity of queried image to pre-processed images, such as recall, precision, Gaussian and Multi quadric functions, [1], Jacobs' [2], Sampson [3],

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 443–451, 2011.

Q. Tian distances [4], etc. In this paper, we propose to use a spline radial basis function as similarity metric with the principal objective to optimize and simplify evaluation process with respect to discrimination methods, particularly comparing it with high-performance GPCA algorithm. Usually, discrimination methods require specific organization or classification of image repository used as collection for VIR. Applying the proposed radial basis function shows, that phase of classifying image collection is not required. In order to test and evaluate the proposed approach the VIR framework called as Image Data Segmentation with Spline (IDSS) has been designed. IDSS retrieves images from a standard collection as COIL-100 using spline radial basis function as similarity metric. [5]. The obtained results are compared with other similarity metrics applied for quantitative evaluation of performance of IDSS framework. Thus, the Gaussian and Multi quadric functions have been used for comparison, which provide lower percentage of similarity grades obtained during matching process of queried and preprocessed images. The analysis of spline similarity behavior is evaluated in Section 2 and description of designed IDSS framework is explained in Section 3. The discussion of preliminary results and concluding remarks are available in Sections 4 and 5 respectively.

2 The Spline Similarity Metric

The image descriptor based on RGB color space may be represented by three dimensions (x,y,z) of Cartesian system of coordinate. After image feature extraction based on color descriptor, interpolation process profiled on radial basis function is applied. This solution provides a set of coefficients that allow to retrieve the most similar images to user query from any image collections ad hoc and at fly. The coefficients obtained from this solution provide automatic retrieval of images with a high degree of similarity avoiding image coefficient adjustment each time a new collection is handled. This advantage of radial basis function as a similarity metric makes the proposed approach invariant to used standard image collection in contrast to other metrics which require previous preprocessing and classification of data in image repositories.

The similarity measure implemented in IDSS system is a radial basis function. The radial basis function (RBF) is based on regression models using linear combinations of coefficients obtained from relevant features of images that depend only on the radial distance (typically Euclidean distance) from a center between a query image and the potentially most similar image at that query [6], [7], [8]. The RBF was introduced for the purpose of a *well-posed* interpolation problem [9] and may be described as follows: First, the surrogate model used in the RBF method is defined. Given a set of input points $x_1, ..., x_n \in \Omega$ with known function values $F_i = f(x_i), i = 1, ..., n$, the radial basis function interpolant s_n has the form

$$s_n(x) = \sum_{i=1}^{n} \lambda_i \phi(\| x - x_i \|) + p(x) \qquad (1)$$

where $\| \ . \ \|$ is the Euclidean norm, $\lambda_1, ..., \lambda_n \in R$ and p is in $\prod_m^d$ (the space of polynomials in d variables of degree less than or equal to m). Where x belongs to the samples set defined as a real number (RGB values for our case). The visual image retrieval area requires novel approaches for searching process. RBF allows efficient matching between images overcoming disadvantages of frequently used similarity metrics as an ordinary euclidean or a Mehalanobis distance only.

Table 1. Different choices of radial basis functions

RBF	$\phi(r) > 0$	$p(x)$	$m_\phi = degree(p(x))$
Thin plate spline	$r^2 logr$	$a^T \bullet x + b$	1
Multi quadric	$\sqrt{r^2 + \gamma^2}, \gamma > 0$	b	0
Gaussian	$e^{-\gamma r^2}, \gamma > 0$	0	-1

Common choices of radial basis functions ϕ and the corresponding polynomial $p(x)$ and minimal polynomial degree m_ϕ are given in Table 1.

The system of linear equations is solved and a set of coefficients previously computed in Table 1 are obtained, that will be used for extracting the surface functions. The surfaces of Spline RBF previously detailed with RGB image data are sketched in Figure 1.

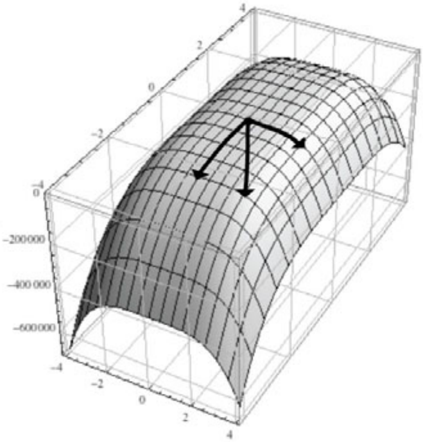


Fig. 1. Representation of RGB vectors of image sets with a thin plate splines surface function

The color descriptor implemented in RGB color space are the coordinates of the Cartesian system of three dimensions (x, y, z). After feature extraction is computed the interpolation process profiled on radial basis function is applied.

This solution provides coefficients that allow to retrieve the most similar images at a user query from any image collections ad hoc at fly. The coefficients obtained form this solution will aid in the retrieval process of images with a high degree of similarity, according to query avoiding adjust coefficients each time a new collection be handled, with an automatic computation achieved. The automatic computation is important in the RBF approach despite other similarity metrics as Jacobs measure that requires adjustments that depends of the collection data.

3 Segmentation Data Modeling System

The process of image data segmentation with splines using designed IDSS framework may be described by two phasesas follows. Phase 1: (a) a data set is provided by standard data base, for example, by COIL-100; (b) a color descriptor is computed extracting RGB feature vectors for analyzed images, (c) a spline RBF is applied as similarity metric computing a set of coefficients (data set) for next phase. Phase 2: (a) to the obtained data set GPCA algorithm is applied, (b) the GPCA algorithm accomplishes classification of data set, (c) the final image classification is provided computing a grade of similarity for particular image retrieval. The block diagram of the IDSS framework is shown in Figure 2. It is important to conclude, that the proposed approach for image retrieval implemented by IDSS framework has two phases. In the first phase, an image retrieval based on RGB color descriptor and splines RBF is provided. The second phase is applied if classification of image collection is required that may be provided by robust GPCA algorithm.

4 Preliminary Results

In order to assess the relevance of the spline radial basis function in the retrieval process, we prepared a training set of 25 image queries. This set, in absence of classification phase, is the input to the first phase of image retrieval by IDSS framework. Table 2 exhibits the behavior of splines during image retrieval process in comparison to Multi quadric and Gaussian RBF's. Table 2 reveals that performance of the approach with spline is higher than Multi quadric or Gaussian RBF functions have. For example, in the provided tests the correct image retrieval reached up to 16 in 25 queries.

The second phase used for image classification is implemented by IDSS framework in the following way: 477 images from the COIL-100 collection previously classified by RGB descriptors are used as input of the GPCA algorithm with randomly basis computation. The visual results are illustrated in Figure 3. The parameters used in the experimental stage have 0.60 level of noise and 0.60 of outliers percentage (details about outliers can be found in [9].

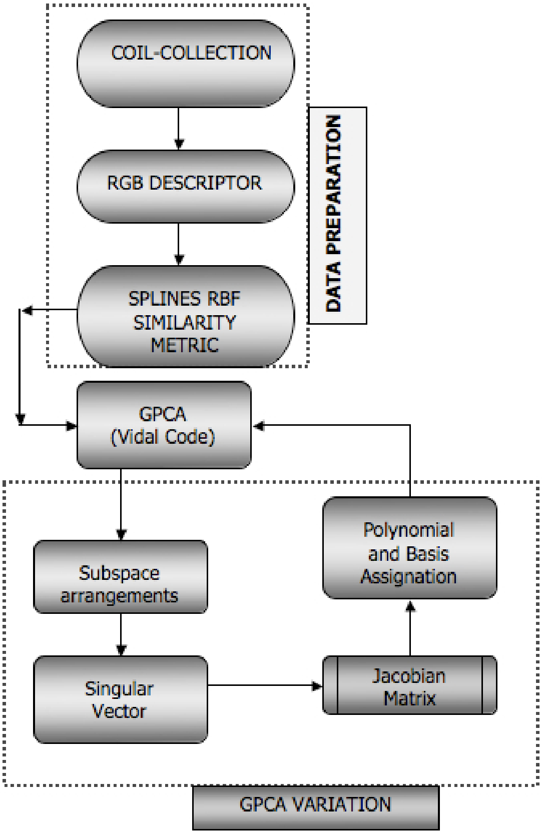


Fig. 2. Sketch of Segmentation Data Image with Splines (IDSS) system

Table 2. Absolute errors for different radial basis functions applied to 25 group samples for queried images from **COIL-100** where * cases involve that the search process was incorrectly retrieved

COIL-100	RBF Function		
Image	Multi quadric	Spline	Gaussian
truck	*	0.000188	*
van	*	*	*
wood spoon	*	*	*
boat	0.038473	0.005234	0.027014
duck	*	0.004284	*
strawberry	*	0.001076	*
butter	*	0.008364	*
kitty	*	0.131610	0.065962
wood2	0.065708	*	*
boat2	*	*	*
onion	*	0.000030	*
tomato	*	0.000696	*
white jar	0.044470	*	0.053097
wood cup	*	0.002623	*
tape	0.012431	0.002677	*
oral	0.041791	*	0.042492
yellow b.	*	0.017720	*
green gum	*	Indet	*
white jar	*	*	*
cigarette	*	0.015038	*
yellow car	*	*	*
red spice	*	0.002924	*
mum	0.007979	0.000338	0.003950
pink s.	*	0.000099	*
white s.	*	0.000815	*

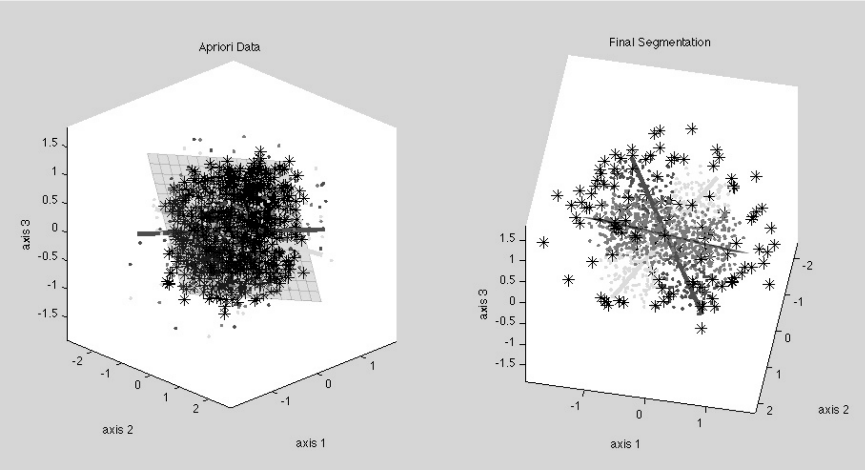


Fig. 3. Left graph: a priori data set of coil image collection. Right graph: Final segmentation with GPCA Robust with three subspace arrangements for each classification of 3 data groups with 477 images.

Table 3 shows the computation of error for each RBF function.

Table 3. Error obtained with Multi quadric, Gaussian and Spline radial basis functions for IDSS with 25 queries, where Spline RBF obtains lower average error value

Function	Retrieval	Error
Multi quadric RBF	6/25	0.035142
Gaussian RBF	5/25	0.038503
Spline RBF	16/25	0.019135

Table 4 reveals that the proposed approach is higher in results in comparison with other metrics as Euclidean distance. The comparisons with some other methods in literature are provided in Table 5 where IDSS is one of the best methods used in image classification in comparison with RGPCA discrimination method.

Table 4. Comparisons of percentages between IDSS with classification phase and without it

Retrieval	Classification	Percentage
RBF	RGPCA	65
RBF	——	88
Euclidean	RGPCA	85
Euclidean	——	80

Table 5. Comparisons of percentages between IDSS and other systems

Author	Systems	Method	Percentage
[10]	IDSS	RBF	88
[6]	REDNEW	WAVELET	65 – 85
[1]	LEPIDOPTERA	2STF	77 – 93
[11]	CIRES	COLOR	58 – 87

5 Conclusions

From these preliminary results, we conclude that the RGB descriptor obviously is not the best choice to classify or to retrieve images, but applying of this approach shows that a previous classification of the data set is not essentially required phase in typical visual information retrieval systems. The second phase of image classification tests with designed IDSS framework uses robust GPCA algorithm because it provides one of the best discrimination method for analysis of random data sets. (see [12], [13], [14], [15], [16] and [17]). However, the application of the second phase is not so required because using the proposed spline radial basis function with the simples RGB feature descriptor provides satisfactory results during image retrieval. Future works suggest applying and evaluating the behavior of approach with different descriptors, for example, descriptors of shape, region or texture. The performance of approach may be improved by analysis of varying size groups of image data and extension of used constrained parameters in Multi quadric RBF function. The main objective of this work is to disseminate the fact that visual information retrieval systems does not requires a previous classification of images in used standard collection if spline RBF is applied as similarity metric. The notable contribution of this research is to reveal that computing of image matching using spline radial basis function is similar to the results obtained by robust discrimination methods, for example, by well-known high performance GPCA algorithm usually used for image segmentation and classification in visual information retrieval applications.

References

1. Starostenko, O., Flores-Pulido, L., Rosas-Romero, R., Alarcon-Aquino, V., Sergiyenko, O., Tyrsa, V.: Shape Indexing and Semantic Image Retrieval Based on Ontological Descriptions. In: Luis, E.S., Hugo, J.E. (eds.) Automatic Image Annotation and Retrieval, vol. 1(1), pp. 58–73 (2011)
2. Portilla Flores, E.A., Vallejo, L.O., Flores Pulido, L., Molina Vilchis, M.A., Acedo Arias, M.A.: Evolución Diferencial para Recuperación de Información Visual (EDRIV). In: SOMI XXIII Congreso de la Sociedad Mexicana de Instrumentación, Xalapa Veracruz, Octubre del (2008)
3. Ma, Y., Yang, A.Y., Derksen, H., Fossum, R.: Estimation of Supspace Arrangements with Applications in Modeling and Segmenting Mixed Data. SIAM Review, Society for Industrial and Applied Mathematics 50(3), 413–458 (2008)

4. Flores Pulido, L., Estrada Cruz, W.E., Chavez Aragon, J.A.: An Image Retrieval System based on Feature Extraction for machine vision using three Similarity Metrics. In: Congreso Internacional Cars & Fof 2007, Universidad Militar De Nueva Granada, Bogotá Colombia, Agosto de 2007, 23rd ISPE International Conference On Cad/Cam, Robotics & Factories Of The Future (2007)
5. Nayar, S., Murase, H.: Technical Report: Columbia Object Image Library (COIL-100), Department of Computer Science, Columbia University, New York, U.S, p. 6 (February 1996)
6. Flores-Pulido, L., Chávez Aragón, J.A., Álvarez Ochoa, L.: Student Debate of PHAROS Summer School, PHAROS Summer School, Chaired by prof. Miki Haseyama, School of Information Science and Technology, Hokkaido University, Hokkaido University Global COE Program Editors, Como Italy (June 2009)
7. Fasshauer, G.E.: Meshfree Approximation Methods with Matlab. Interdisciplinary Mathematical Sciences, vol. 6. World Scientific, Singapore (2007)
8. Regis, R.G., Shoemaker, C.A.: Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions. J. Glob. Optim. 31, 153–171 (2005)
9. Iske, A.: Lambda-Calculus and Computer Science Theory. Lecture Notes in Computational Science and Engineering, vol. 37. Springer, Heidelberg (1975) ISBN 3-540-20479-2
10. Flores-Pulido, L., Starostenko, O., Rodriguez-Gomez, G., Alarcón-Aquino, V.: Segmentation Data Modeling for Image Retrieval System. In: Proceedings of LANMR Workshop in CEUR Workshop Proceedings 2010, CEUR-WS.org, publication service of Sun SITE Central Europe operated under the umbrella of RWTH Aachen University with the support of Tilburg University. CEUR-WS.org, a publication series by Deutsche Bibliothek (2010), ISSN 1613-0073
11. Iqbal, Q.: Content Based Image Retrieval System, Univ. of Texas at Austin, Ph.D (2007), http://amazonece.utexas.edu/~qasim/research.htm
12. Björner, A., Peeva, I., Sidman, J.: Subspace arrangements defined by products of linear forms. J. London Math. Soc. 71(2), 273–288 (2005)
13. Derksen, H.H.: Hilbert Series of Subspace Arrangements, preprint, arXiv.org (2005), http://arxiv.org/abs/math/0510584
14. Vidal, R., Ma, Y., Piazzi, J.: A New GPCA Algorithm for Clustering Subspaces by Fitting, Differentiating and Dividing Polynomials. In: CVPR, pp. 510–551 (2004)
15. Donoho, D.: Compressed sensing. IEEE Transactions on Information Theory, 1289–1306 (April 2006)
16. Sugaya, Y., Kanatani, K.: Outliers removal for motion tracking by subspace separation. IEICE Trans. Inform. Systems E86-D, 1095–1102 (2003)
17. Huang, K., Wagner, A., Ma, Y.: Identification of hybrid linear time-invariant systems via subspace embedding and segmentation. In: Proceedings of the IEEE Conference on Decision and Control, vol. 3, pp. 3227–3234 (2004)

A Comparative Review of Two-Pass Connected Component Labeling Algorithms

Uriel H. Hernandez-Belmonte, Victor Ayala-Ramirez, and
Raul E. Sanchez-Yanez

Universidad de Guanajuato DICIS, Carr. Salamanca-Valle Km. 3.5+1.8, Com. Palo Blanco, 36700 Salamanca, Mexico
hailehb@laviria.org,{ayalav,sanchezy}@ugto.mx

Abstract. In this paper, we show a comparative review of Connected Component Labeling (CCL) methods, focused in two-pass variants, including their elements and implementation issues. We analyze the main elements used by these CCL algorithms and their importance for the performance of the methods using them. We present some experiments using a complex image set and evaluating the performance of each algorithm under analysis.

1 Introduction

One of the most used methods in image processing applications is the Connected Component Labeling (CCL). The CCL procedure assigns a unique label to a set of connected target pixels in a binary image. A binary image is an image where all the pixels only take one two values. One value is used for the foreground (e.g, black pixels) and the other for the background pixels (e.g. white pixels). We say that two pixels are connected if there exists a path between them. Each component is composed of a set of pixels that share a common binary property. This image property can be the result of a segmentation process or of any other thresholding method.

One of the main goals of CCL methods in the literature has been to reduce its computational cost. According to [9], we can classify CCL algorithms in three groups:

1. Multi- pass algorithms, where the image is scanned several times. The work by [6] is an example.
2. Two pass algorithms, where the image is scanned only two times. The method proposed by [7] is an example of this class.
3. Tracing-type algorithms, where the connectivity is represented in an implicit way. See for example, [1].

Label equivalence handling is a crucial issue for the development of CCL methods. That is why we will compare the performance of two algorithms, to know: 1) the Union-Find approach, originally proposed by [5] and further enhanced

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 452–462, 2011.

by [13] and 2) the Connection-List approach proposed recently by [8]. Each algorithm uses a different approach to represent the labels and in consequence how they handle equivalence among them. Both algorithm were implemented and they were tested with a set of inputs and we have compared their execution time.

Rest of this paper is organized as follows: Section 2 is a review of main approaches to CCL problem. In Section 3, we describe the elements of two pass CCL algorithms. We review the main implementation for these methods in Section 4. Section 5 shows the test protocol used to compare both methods and the results obtained from its execution. We present also there explanations about these results. Finally, Section 6 presents the insights gained from the comparative review of the $Union - Find$ and $Connection - List$ approaches for solving the CCL problem.

2 Connected Component Labeling

Pioneering works on CCL include those by [10] and [6]. They have used multiple scans of the images over neighborhoods defined by a connectivity mask. In these early works, label propagation strategies were not optimal.

Other approaches to solve the CCL problem use binary trees and quad-trees [11] [4] to represent the input image. Using these data structures, the image is analyzed in a recursive way by partitioning in sub-divisions until all the leaf nodes of the tree represent a region in the image composed of pixels of the same type. After that, an adjacency analysis is done to associate leaf nodes to the connected components in the image.

An interesting approach to handle equivalence labeling that let us to get additional information during the labeling procedure was proposed by [2]. This algorithm detects simultaneously the connected components and the external and internal contours for the connected components.

More recent approaches addressing the CCL problem use two pass algorithms [5], [3], [12], [7], [13]. The first scan of the input image serves to assign temporary labels for the target pixels while in the second scan, label equivalences are resolved. To find the correct labels for the target pixels. To perform efficiently label equivalence resolution, we try to minimize the memory cost and the number of operations needed to find the correct label. Another issue is to reduce the number of operations needed to inspect the neighborhood of a pixel under analysis in order to determine if there are conflicts in label assignment.

3 Elements of Two-Pass CCL

3.1 Connectivity Analysis

The connectivity analysis decides what operation needs to be done for the pixel under analysis. The two-pass CCL methods use a connectivity mask that covers a neighborhood of the pixel under analysis. The information of the pixels in

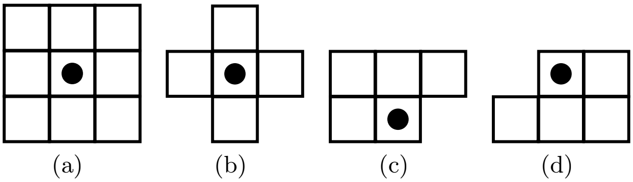


Fig. 1. Typical shapes of neighborhood masks used for connectivity analysis in CCL algorithms, the pixel under analysis is marked on them: (a) 3 × 3 mask, (b) 4-neighbor mask, (c) forward scan mask and (d) backward scan mask

the spatial region covered by the connectivity mask is used as support for the decision. Figure 1 shows some of the most often used connectivity masks.

To exemplify the connectivity analysis, let us consider a 3 × 3 mask (Fig. 1a), also called eight-connected mask. We consider a binary image I as input and the goal of the connectivity analysis is to assign a label $L(x, y)$ to each foreground pixel. Let m be the count number of component labels, and let initialize $m = 0$. Let us consider $I(x, y)$ the intensity of the pixel under analysis, and $L(x, y)$ the label assigned to that pixel. There are three different cases for the label assignment problem:

i) If $I(x, y) = I_B$, no operation is needed and the pixel is labeled $L(x, y) = L_B$, with L_B being the label assigned to each background pixel.
ii) All the pixels in the neighborhood defined by the connectivity mask are background pixels and the pixel under analysis is a foreground pixel. That is,

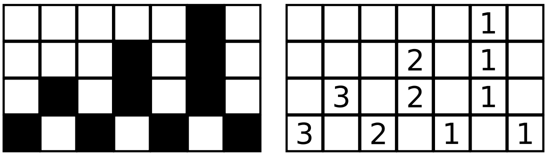
$$(I(x, y) = I_F) \cap (I(x + i, y + j) = I_B) \qquad \forall (i, j) \in M_S \tag{1}$$

with M_S being the spatial support of the connectivity mask. In such a condition, we need to create a new component label.
iii) If there exist different labels already assigned to the pixel in the neighborhood under analysis, we select the label with the lower component index among those appearing in the pixel neighborhood and we assign it to the pixel under analysis. That is,

$$L(x, y) = \min_{(i,j) \in M_S} L(x + i, y + j). \tag{2}$$

3.2 Equivalence Handling

Given the sequential procedure for labeling components in a binary image, after a first scanning of the entire image, target pixels are labeled with a label that could not be the definitive one. Let us take as an example the Finger pattern in Figure 2(a). If we use as connectivity mask an eight-connected mask, we obtain the labeling depicted in Figure 2(b) after the first scanning. The single component in the image is composed of pixels with labels $\{1\}, \{2\}, \{3\}$. The label equivalence handling method is charged of joining all these labels in a single set $\{1, 2, 3\}$, that is finally represented by the lower label index, in the example, $\{1\}$. The second scan exchanges all the temporary pixel labels by the representative label of the set to which the temporary label belongs. In the presented example, all the pixels will be finally assigned a label of $\{1\}$, denoting that there is a single component in the image.



(a) (b)

Fig. 2. Labeling after the first pass of a CCL algorithm on a Finger pattern

The algorithm for label equivalence handling has to implement the following operations:

New label creation, performed when there are no foreground pixels in the neighborhood under analysis. It needs also to record what labels have already been created in order to avoid using a previously created label.

Joining two sets, when the connectivity analysis results in the detection of two or more labels in the neighborhood under analysis, we need to join these sets into a single set.

Determining the representative label of a set, when we need to represent a set of labels by a single representative label. That operation is needed to exchange temporary labels by the definitive ones.

Each CCL algorithm implements these functions in a specific way. The label equivalence handling algorithms need to define data structures suitable to represent equivalence among labels. The previously described operations work on the information stored on these data structures to perform label equivalence handling.

3.3 Union-Find Labeling

The Union-Find algorithm uses two main elements: a vector of nodes called *Parent*, used to store equivalence information; and the operations needed to handle the information on it. Each node of the parent vector consists of a label and a pointer to its root node. The size of the *Parent* vector depends of the maximal count of different labels that could be needed to label the connected components in the image. In fact, for a binary image I with $N \times M$ pixels, the maximum number of labels is $(N \times M)/4$. We initialize all the elements of the *Parent* vector as root nodes. After that, they point to themselves and when a new label is created there are no operations to be done on the *Parent* vector. Each element of the *Parent* vector is assigned with a label corresponding to its index in the vector. That is, $parent[m] = m$, $m \in \{0, 1, 2, \ldots (N \times M)/4\}$.

For this vector, we need three operations to handle label equivalence:

i) A new node generation.
ii) To join two trees (Union).
iii) To find the root node associated to another node (Find).

The Union procedure is invoked when there are more than one label in the neighborhood defined by the connectivity mask. For each pair of labels, we look for the root nodes of both labels, we choose the minimum value from them and we assign then the chosen label as the parent node of both input nodes.

The Find operation is needed to trace back the root node of a tree of equivalent labels. This operation is divided in two steps: finding the root node and, modifying all the nodes in the path to the root node to point to it as its parent node.

3.4 Connection-List

The Connection-List method represents the label equivalence using a set $S(r)$, also called *equivalent label list*, where r is known as the representative label of the set and r takes the smallest label in the set. The Connection-List approach also uses a table R called *representative label table*, to store the relationships in the label sets and their representative label, that is:

$$\forall l \in S(r), R(l) = r \tag{3}$$

Using this idea, all labels in an equivalent label list set are part of the same connected component, and the representative label is the label to be used for the component.

To implement the Connection-List approach, it is necessary to use three arrays. These arrays are called *next*, *tail* and *rtable*. The *next* array is used to represent the label next to the previous label in the list. If $next[l] = -1$, label l is the tail label of the list. The *tail* array is used to indicate the tail label of a provisional label set. The *rtable* is used to store the representative label list.

The operations needed to handle label equivalence used by the Connection-List method are:

i) *new equivalent list*,
ii) $merge(u, v)$, and
iii) $resolve(x, y)$.

The first operation is used to create a new equivalent label list. The second operation joins two provisional labels (u, v) in the same label list, where u and v are the representative labels of the components being merged and $u < v$. The third operation is for resolving the equivalence of two provisional labels x and y. This operation does not requiere any particular ordering for the x and y labels. Resolve operation is invoked without needing to know if x and y are the representative labels of the component being merged.

4 Labeling Procedure for Two-Pass CCL Methods

The labeling procedure follows the same steps in two-pass CCL algorithms. Firstly, we need to initialize the data structure used for label equivalence handling. In the case of the $Union - Find$ algorithm, we initialize the $Parent$ structure and for the $Connection - List$ approach, we initialize the $next$, $tail$ and $r - table$ arrays. The size of each array mentioned before depends on the input image size. This size is $(N \times M)/4$, where N is the number of rows and M is the number of columns of the input image. In order to simplify the connectivity analysis, a dummy row and a dummy column of background pixels are added both at the top-left and at the bottom-right positions of the image.

The connectivity analysis is done by implementing a lookup table with all the configurations that the neighborhood set can exhibit. We encode the lookup table by giving a particular order (weighting) to each position in the mask. Figure 3 shows an enumerated eight-connected connectivity mask, where $\{a, b, c, d\}$ are the neighborhood of the pixel and k the pixel under analysis.

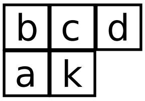


Fig. 3. Enumerated eight-connected connectivity mask

The typical weighting used for the neighborhood pixels $\{a, b, c, d\}$ is $\{a \times 2^3, b \times 2^2, c \times 2^1, d \times 2^0\}$. Adding the pixel values using this weighting, we obtain a number of pattern P that is used as the index to the operation lookup table. The weighting factor can be easily implemented as a binary shift operation to reduce its computational cost.

Table 1 presents the operation lookup table indexed by the pattern P (first column) exhibited by the neighborhood of the pixel under analysis. Columns from 2 to 5 show the configuration of the neighborhood and the sixth column shows the corresponding label. Last two columns present the operations executed by both algorithms, $Union - Find$ and $Connection - List$, respectively.

After the first pass, we will have updated the image $L(x, y)$ with the temporary labels. The second step is performed by traversing the image again and a decision is taken about the definitive label of each pixel. In the case of the $Union - Find$ algorithm, we execute the operation $L(x, y)' = find(L(x, y))$, where $L(x, y)'$ is the image with correct labels. If we use the $Connection - List$ algorithm, we execute the operation $L(x, y)' = R(L(x, y))$ that finds the representative label associated to the temporary label assigned in the first pass. This is implemented by executing $L(x, y)' = rtable[L(x, y)]$. When the second pass is finished, all the pixels in the image have the correct label of the connected component to which they belong.

Table 1. Look-up table operations associated to each-connected mask configuration

P	a	b	c	d	k	*Union-Find*	*Connection-List*
0	0	0	0	0	$m = m + 1$	NOP	NOP
1	0	0	0	1	d	NOP	NOP
2	0	0	1	0	c	NOP	NOP
3	0	0	1	1	d or c	NOP	NOP
4	0	1	0	0	b	NOP	NOP
5	0	1	0	1	b or d	NOP	NOP
6	0	1	1	0	b or c	NOP	NOP
7	0	1	1	1	b, c or d	NOP	NOP
8	1	0	0	0	a	NOP	NOP
9	1	0	0	1	a or d	$Union(a, d)$	$resolve(a, d)$
10	1	0	1	0	a or c	$Union(a, c)$	$resolve(a, c)$
11	1	0	1	1	a, c or d	$Union(a, d)$	$resolve(a, d)$
12	1	1	0	0	a or b	NOP	NOP
13	1	1	0	1	a, b or d	NOP	NOP
14	1	1	1	0	a, b or c	NOP	NOP
15	1	1	1	1	a, b, c or d	NOP	NOP

5 Tests and Results

We have implemented two algorithms: the first one ($Union - Find$) was implemented by following the work of [13] and the second one ($Connection - List$) by using the approach presented in [9]. The tests evaluate the performance of both algorithms with respect to the execution time when two test sets are used as input.

The first test set is composed of images having structured patterns on them. We have tested the algorithm over images constructed using sieve, step, spiral and finger benchmark patterns with varying image sizes. The elementary tile patterns of these images are shown in Figure 4 (a), (b), (c) and (d), respectively.

The second set consists of random noise images of size 512 × 512 with varying occupancy percentage of foreground pixels. We have generated a base of 50 images for each percentage of occupancy from 0 to 100 percent. Figures 5 (a) and (b) are examples of noisy images, where white pixels are considered as background pixels and black pixels are foreground pixels.

Our computer implementations were developed using C language and the gcc 4.0 compiler. The desktop computer for the tests is an Intel Dual Core-based with a processor running at 2.66 GHz and 4 GB RAM memory.

Sieve and Step test patterns. In Figures 6 (a) and (b), we can observe there that $Union - Find$ method overperforms the $Connection - List$ algorithm. That can be explained because $Union - Find$ label creation procedure has a lower computational cost than the similar procedure of the $Connection-List$ approach. When we create a new label in the $Union - Find$ algorithm, we

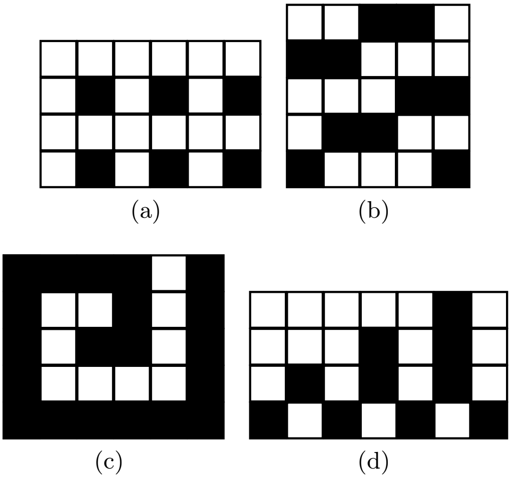


Fig. 4. Elementary patterns of the benchmark images used for performance evaluation of the four methods under analysis: (a) sieve, (b) step, (c) spiral and (d) finger patterns

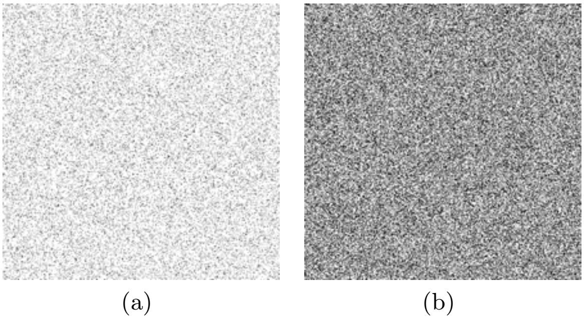


Fig. 5. Two examples of uniform random noise images with varying occupation percentage of the foreground pixels: (a) 10 % and (b) 40 %

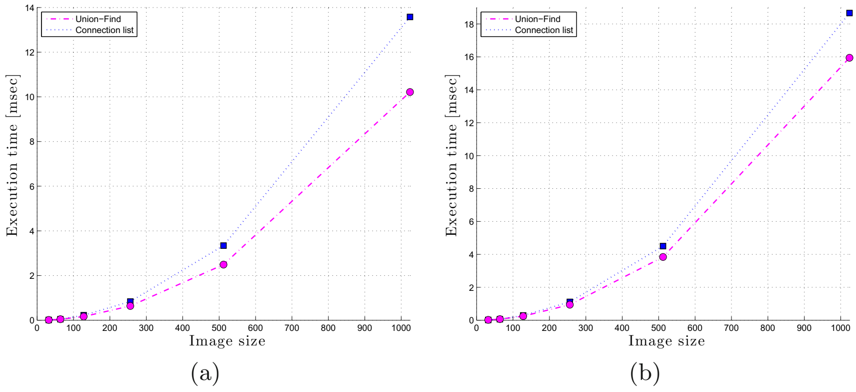
need to create only a new root node. Instead, we need to create a new connected list in the case of the $Connection - List$ method.

We can observe this behavior more clearly in the results for a Sieve pattern input (Figure 6 (a)). Given that all the components in the Sieve pattern are constituted of a single pixel, there is a new label creation procedure for each component in the first scan. In the second scan of this specific pattern, the cost of the $find(l)$ operation is the same than the $R(l)$ because all the nodes detected in the first scan are root nodes.

Spiral and Finger test patterns. The results are shown in Figures 7 (a) and (b) for these two test cases. Under the conditions exhibited by these patterns, the $Connection - List$ algorithm expends less time in removing temporary labels.

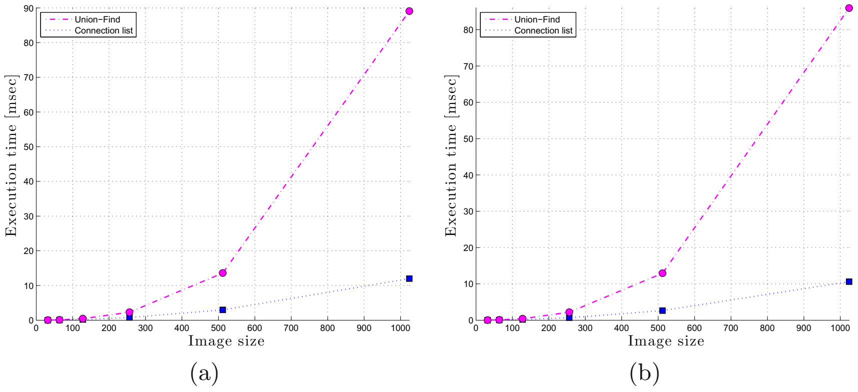
In the case of the $Union - Find$ algorithm, the method needs to review for each foreground pixel in the input images if the pixel label is a root node and if it is not, it needs to search for a root node. That will imply more operations to find the root node than in the case of the $Connection - List$ approach, where with a simple lookup procedure of the array $rtable[l]$ we can get the representative label for each pixel. This lower cost shows to be the reason for the time execution advantage of the $Connection - List$ approach on both test patterns discussed here.

Random noise images. Figure 8 shows the results for the random noise input images. We can observe that the $Connection - List$ algorithm performs better than the $Union - Find$ approach over the most of the range of the occupancy percentage (40 to 100%). Nevertheless, we can observe that in low foreground pixel density images (from 0 to about 30 %), the $Union - Find$ performs better.



(a) (b)

Fig. 6. Comparison of the performance of $Union - Find$ and $Connection - List$ algorithms when (a) Sieve and (b) Step patterns are used as input



(a) (b)

Fig. 7. Comparison of the performance of $Union - Find$ and $Connection - List$ algorithms when (a) Spiral and (b) Finger patterns are used as input

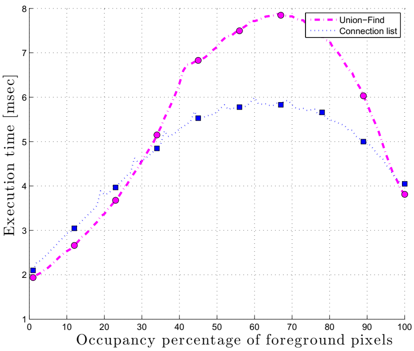


Fig. 8. Performance results of the two methods using uniform random noise images of varying occupancy percentage of the foreground pixel as input (image size is 512 × 512 for all test cases)

6 Conclusions

In this paper, we have shown the importance of the algorithms for label equivalence handling in CCL algorithms by doing a comparative review of two-pass CCL algorithms. We have described the main elements of them. This review has been performed on the $Union-Find$ algorithm and on the $Connection-List$ algorithm.

We have also discussed the connectivity analysis done by using connectivity masks. We discuss what are the considerations to take into account in order to make a decision about the operations to be performed on each pixel under analysis and how we can encode all this information in the operation lookup table.

We have performed tests on both algorithms by using a complex image set as input. We have shown in the tests that both approaches are suitable for the CCL problem but that their usefulness is dependent on the nature of the input image.

In the near future, we will work towards determining automatically what is the more advantageous method to apply for a given input image.

Acknowledgements. Uriel H. Hernandez-Belmonte gratefully acknowledges Mexico's CONACYT for the financial support through the scholarship 329356/229784.

References

1. Chang, F., Chen, C.-J.: A component-labeling algorithm using contour tracing technique. In: International Conference on Document Analysis and Recognition, pp. 741–745 (2003)

2. Chang, F., Chen, C.-J., Lu, C.-J.: A linear-time component-labeling algorithm using contour tracing technique. Computer Vision and Image Understanding 93(2), 206–220 (2004)
3. Di Stefano, L., Bulgarelli, A.: A simple and efficient connected components labeling algorithm. In: International Conference on Image Analysis and Processing, pp. 322–327 (1999)
4. Dillencourt, M.B., Samet, H.: A general approach to connected-component labeling for arbitrary image representations. Journal of the ACM 39(2), 253–280 (1992)
5. Fiorio, C., Gustedt, J.: Two linear time Union Find strategies for image processing. Theoretical Computer Science 154(2), 165–181 (1996)
6. Haralick, R.M.: Some Neighborhood Operations. In: Real Time/Parallel Computing Image Analysis, pp. 11–35. Plenum Press (1981)
7. He, L., Chao, Y., Suzuki, K.: A linear-time two-scan labeling algorithm. In: International Conference on Image Processing, pp. 241–244 (2007)
8. He, L., Chao, Y., Suzuki, K.: A run-based two-scan labeling algorithm. IEEE Transactions on Image Processing 17(5), 749–756 (2008)
9. He, L., Chao, Y., Suzuki, K., Wu, K.: Fast connected-component labeling. Pattern Recognition 42(9), 1977–1987 (2009)
10. Rosenfeld, A., Pfaltz, J.L.: Sequential operations in digital picture processing. Journal of the ACM 13(4), 471–494 (1966)
11. Wang, X., Davis, W.A.: Connected component labeling using modified linear quadtrees. In: Graphics Interface 1986, pp. 235–240 (1986)
12. Wu, K., Otoo, E., Suzuki, K.: Two strategies to speed up connected component labeling algorithms. Tech Report LBNL-59102 (2005)
13. Wu, K., Otoo, E., Suzuki, K.: Optimizing two-pass connected-component labeling algorithms. Pattern Analysis and Applications 12, 117–135 (2009)

A Modification of the Mumford-Shah Functional for Segmentation of Digital Images with Fractal Objects

Carlos Guillén Galván, Daniel Valdés Amaro, and Jesus Uriarte Adrián

Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
Puebla, México, 72570
{cguillen,daniel.valdes}@cs.buap.mx, vasco_25@hotmail.com

Abstract. In this paper we revisit the Mumford-Shah functional, one of the most studied variational approaches to image segmentation. The contribution of this work is to propose a modification of the Mumford-Shah functional that includes Fractal Analysis to improve the segmentation of images with fractal or semi-fractal objects. Here we show how the fractal dimension is calculated and embedded in the functional minimization computation to drive the algorithm to use both, changes in the image intensities and the fractal characteristics of the objects, to obtain a more suitable segmentation. Experimental results confirm that the proposed modification improves the quality of the segmentation in images with fractal objects or semi fractal such as medical images.

1 Introduction

Many techniques that improve the Mumford-Shah functional have been proposed. Song and Chain [1] perform image segmentation using a multiscale method, where instead of solving the corresponding Euler-Lagrange equation, a direct computation of the energy is performed and then, changes in it are tested when a point change inside or outside of the level set is done. Chan and Esedoglu [2] proposed a segmentation method that works faster by allowing the algorithm to operate at more scales (number of pixels) than that of a single pixel, and showed how some of the premature steady states can be avoided to improve the original Song and Chan algorithm. An and Chen [3] presented two based variational partial differential equation (PDE) models for image segmentation. The first model uses a modified piecewise constant Mumford-Shah model and the second one is obtained using a prior shape information and region intensity values. In [4] a nonlinear smooth constraint function is introduced that can induce edge-preserving regularization to facilitate the coupled image segmentation. Finally, Pan *et al* [5] implemented a technique that uses the functional to perform a bottom-up hierarchical segmentation, merging the competition of regions and the Mumford-Shah functional.

As well, work has been done in segmentation using fractal analysis. For example, Delrieux [6] proposes a segmentation approach adequate for noisy images that instead of detecting local changes in the brightness level, a previous step of local fractal dimension is applied. Then, a histogram classification of this attribute of the image allows an adequate automatic threshold detection for the segmentation. In [7] we can find a

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 463–473, 2011.

comparison of two applications, the first one uses fractal encoding as a mean to identify anomalous regions in an image, and the second application uses fractal encoding to redetect and redefine image regions. The work of Vuduc [8] introduces a local operator that can measure the fractal dimension of each image pixel, and for such purposes, then using histograms of the computed fractal dimensions of all the pixels the image is recolored in bins producing a segmented image.

As none of the modifications of the Mumford-Shah Functional involves the use of the fractal dimension, and none of the segmentation methods that use the fractal dimension uses it to characterize the objects to be segmented, in this paper we propose a modification of the Mumford-Shah functional that includes Fractal Analysis to improve the segmentation of images with fractal or semi-fractal objects. This requires an analysis of the impact of the fractal dimension in the Mumford and Shah model, to first obtain a continuous model of adaptation and secondly get a discretized model for the implementation. Important to note that since Mandelbrot in his work *The Fractal Geometry of Nature* [9] proved that fractional dimensions were useful in the characterization of natural phenomena, many studies have applied the framework of fractal geometry. So too is the case of biology and medicine, where the complexity in the structures of living creatures resembles the principles of fractal geometry. Due to the complexity of the geometry of human body structures it is plausible to assume that for instance, the brain, lungs, veins or vessels possesses a self-similar structure. To explain this, different studies have analysed such structures [10,11,12,13,14]. Hence, we consider this type of objects as semi-fractal and suitable to experiment with.

2 The Mumford-Shah Functional

Mumford-Shah (MS) functional is an important variational model in image analysis. It minimizes a functional involving a piecewise smooth representation of a given image and penalises the Hausdorff measure of the set of discontinuities. This results in simultaneous linear restoration and segmentation [15].

In the Mumford-Shah approach, the aim is to find an image close to the original one, that is composed of several regions with near constant intensities. This function involves curvature measurement as a geometric quantity. This problem can be located in the so-called free discontinuity problems, terminology introduced by De Giorgi [16].

The functional is based on minimizing [15]:

$$F(u,K) = \int_{\Omega - K} (u - u_0)^2 dx + \alpha \int_{\Omega - K} |\nabla u|^2 dx + \beta \int_K d\sigma \tag{1}$$

where Ω is an open set of $\mathbb{R}^N$ with N = 2,3; u_0(x) is the initial image, α and β are nonnegative constants ; with u_0 : Ω →[0,1] and K belongs to Ω.

The difficulty of this technique consists in the fact that involves two unknowns, the intensity function and the set K of discontinuities. Equation 1 is composed by three terms, the first one indicates the level of accuracy of the segmentation, and denotes how well do the smooth image u approximates to the original image u_0. The second term is the smoothness of the intensities, that must be small if u changes slowly in the different colour regions. And finally, the third term is the length measurement of all edges, that must be small to prevent the edges to occupy the entire image [17].

Different methods to approximate the Mumford-Shah's functional can be found in [15]. This work is based on the approximation of the finite differences scheme since it presents an ideal framework to find the solution of the Mumford-Shah's functional as a curve evolution problem, where the aim is to find the best curves that meet the functional criteria to this approximation. For details see [15].

3 Fractal Analysis and the Mumford-Shah Functional

In this section we describe the method to calculate the Fractal Dimension and how this is incorporated in the minimization frame of the Mumford-Shah Functional.

3.1 Estimating Fractal Dimension

The fractal dimension of a curve estimates how much space it occupies relative to its length. For a planar curve, it can be calculated by the number, $N(\delta)$, of area elements, δ, needed to cover the curve. The estimation of the curve length is then $N(\delta)\delta$. The Hausdorff-Besicovitch dimension D of a curve is defined as some measure $M_d(\delta)$ for which $N(\delta)\delta^d$ abruptly changes from zero to infinity [18]. Since D is often finite, it can be shown that for sufficiently small δ

$$D = -\frac{logN(\delta)}{log(\delta)}. \tag{2}$$

For practical purposes, the *Box Counting dimension* was used in this work because it is relatively easy to calculate mathematically and can be estimated empirically. Let **F** be any non-empty bounded subset of $\mathbb{R}^n$ and let $N_\delta(\mathbf{F})$ be the smallest number of sets of diameter at most δ which can cover **F**. If the lower and upper box-counting dimensions of **F** are equal, then we refer to the common value as the *box-counting dimension* or *box dimension* of **F** [19]:

$$dim_B\mathbf{F} = \lim_{\delta\to 0}\frac{log\ N_\delta(\mathbf{F})}{-log\ \delta} \tag{3}$$

In other words, to obtain the box dimension of a plane set **F** we can consider drawing a mesh or a set of boxes of side length δ and then count the number $N_\delta(\mathbf{F})$ that overlap the set for various sizes of δ (hence the name 'box counting'). Thus, we may determine the fractal dimension by finding the slope of $log\ N_\delta(\mathbf{F})$ plotted as a function of $-log\ \delta$ [18].

3.2 The Mumford-Shah Functional Revisited

In [15] authors present an active contour model similar to the binary segmentation that uses minimum energy principles to accomplish the task. The main idea of this model is to consider the information not only in the boundaries but in the regions as well. To be more precise, the aim is to find a better approximation of the image u_0 through a set of regions with two different intensities i_1 and i_2. One of the regions represents the objects to be detected (inside(C)) and the other one, (outside(C)), represents the background.

Therefore the model is reduced to the following minimization problem:

$$inf_{i_1,i_2,C}\, F(i_1,i_2,C) \tag{4}$$

where

$$F(i_1,i_2,C)=\mu|C|+\int_{inside(C)}|u_0-i_1|^2dx+\int_{outside(C)}|u_0-i_2|^2dx \tag{5}$$

with μ positive. In [20] the relationship between this model and the Mumford-Shah Functional is established for the segmentation as follows:

$$F^{MS}(u,C)=\mu|C|+\lambda\int_{\Omega}|u-u_0|^2dx+\int_{\Omega\setminus C}|\nabla u|^2dx \tag{6}$$

where μ and λ are positive parameters and the solution image u obtained by minimizing this functional is composed by smooth regions R_i and with sharp boundaries denoted by C.

The reduced form of this problem is simply the restriction of F^{MS} to piecewise constant functions u, this is, $u(x)=c_i$, with c_i constant on each connected component R_i of $\Omega\setminus C$. Hence, the constants c_i are the averages of u_0 on each region R_i.

As mentioned before, the aim of this work is to enhance the segmentation quality for images containing fractal objects. Analysing the reduced form of the restriction of F^{MS} is necessary to incorporate this model to the fractal nature of the image. To achieve this, the fractal dimension of an image fd is integrated in the iterative process of the model. Following the same idea of minimal energy we propose the addition of a restriction to the minimizing function that considers the fractal dimension of the input image. Such restriction is given by the minimum difference between the fractal dimension of u_0 (FD) and the one of u (fd). This additional restriction is considered in the image accuracy term, so in the simplified functional model we add the term:

$$\beta|FD-fd|+1 \tag{7}$$

where β is a constant in the order of 10^6, FD is the estimated fractal dimension of the input image u_0 and fd is the fractal dimension of u. The calculations of the fractal dimensions are performed using the method explained in the previous section.

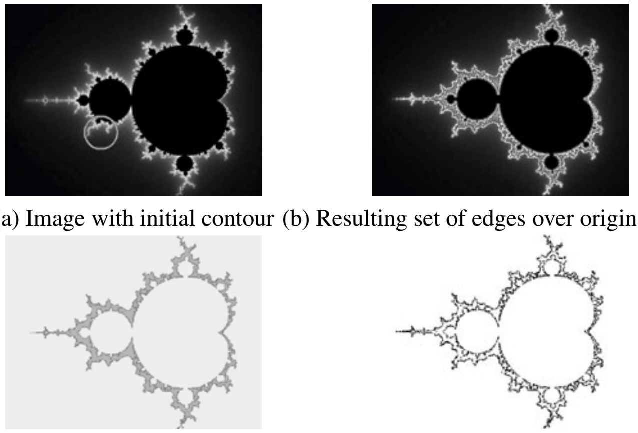
The impact of this restriction is shown in equation 5 so that in the active contours algorithms we must replace the original equation with the following one:

$$F(i_1,i_2,C)=\mu|C|+\int_{inside(C)}|u_0-i_1|^2dx+\int_{outside(C)}|u_0-i_2|^2dx+(\beta|FD-fd|+1) \tag{8}$$

In the following section we show the results of the modified functional.

4 Experimental Results

To evaluate the effectiveness of the proposed model, we first show results on true fractal images. The first tested image corresponds to the Mandelbrot set, and in figure 1 we present the segmentation using the standard Mumford-Shah functional implementation, whereas in figure 2 results for our modified algorithm are presented. For both tests, 2000 iterations were used. For details on the initial parameters of the algorithms see table 1.



(a) Image with initial contour (b) Resulting set of edges over original image

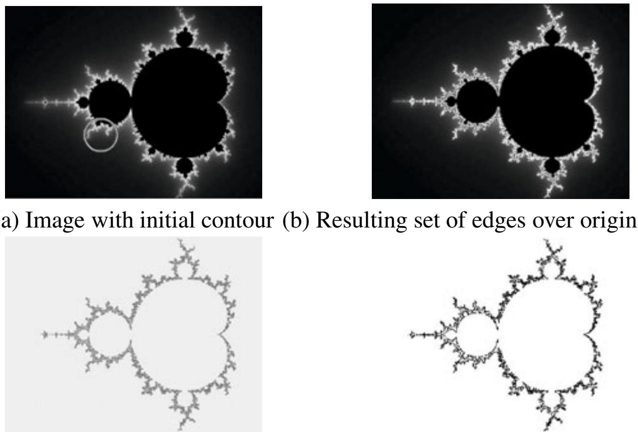
(c) Resulting set of regions (d) Resulting set of edges

Fig. 1. Segmentation of the Mandelbrot set using the standard Mumford-Shah functional implementation

For our second experiment a Julia set image was used. Again, in figure 3 we present the segmentation using the standard Mumford-Shah functional implementation and in figure 4 results for our modified algorithm. 3000 iterations were used for both tests. For details on the initial parameters of the algorithms see table 1.

Next experiments used a simulated human brain MR image obtained from the BrainWeb Project (http://mouldy.bic.mni.mcgill.ca/brainweb/). As mentioned before, the human brain exhibits fractal characteristics, so this type of images are adequate for our purposes. Figure 5 presents the segmentation using the standard Mumford-Shah functional implementation and figure 6 illustrates results for our modified algorithm. In this experiment we ran the algorithm for 1000 iterations. For more details on the initial parameters of the algorithms see table 1.

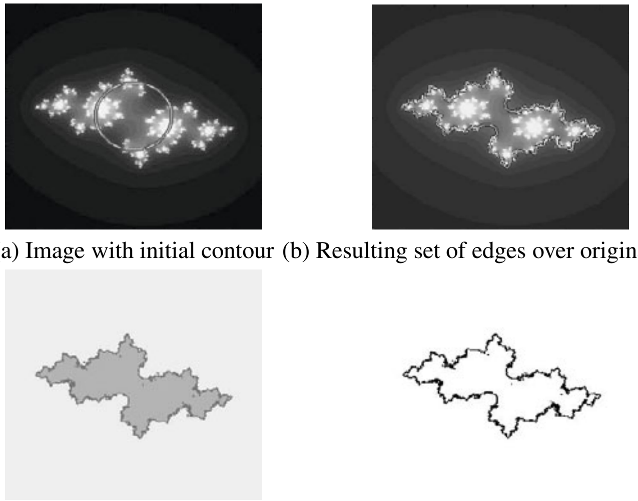
Finally the last experiment uses a human eye vessel image. Vessels and their branches, exhibit much variability with artifacts and low contrast, making segmentation very difficult. Moreover, this kind of images are suitable due to the fractal nature of such structures. Figure 7 depicts the segmentation results using the standard Mumford-Shah functional implementation and in figure 8 we can find results for the modified algorithm. Again 1000 iterations were used, so for details on the initial parameters of the algorithms see table 1.



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

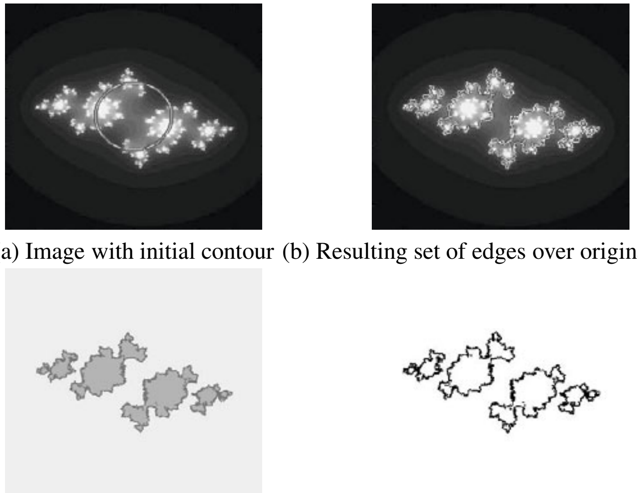
Fig. 2. Segmentation of the Mandelbrot set using the modified Mumford-Shah functional implementation



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

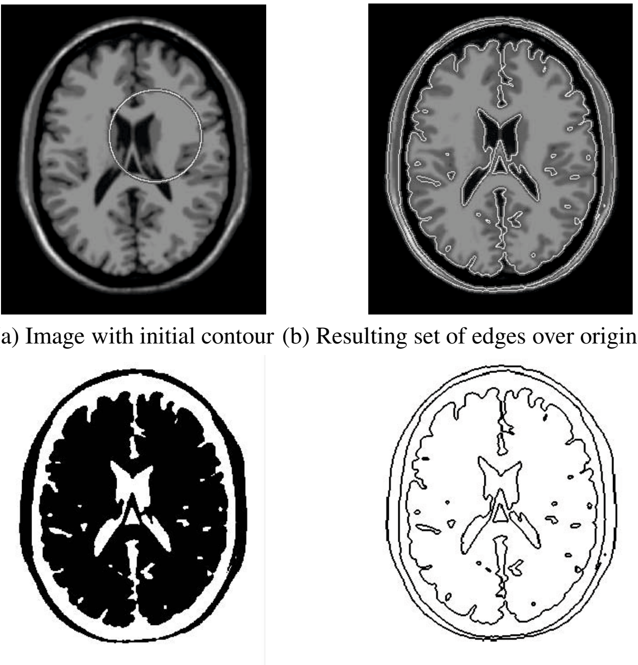
Fig. 3. Segmentation of the Julia set using the standard Mumford-Shah functional implementation



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

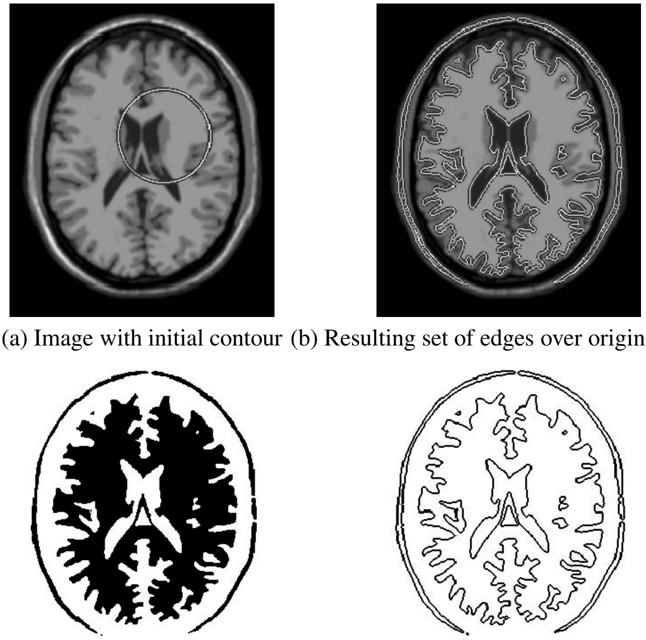
Fig. 4. Segmentation of the Julia set using the modified Mumford-Shah functional implementation



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

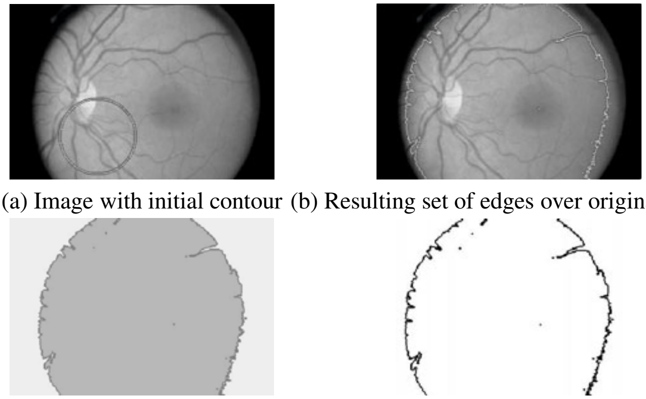
Fig. 5. Segmentation of a human brain MR image using the standard Mumford-Shah functional implementation



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

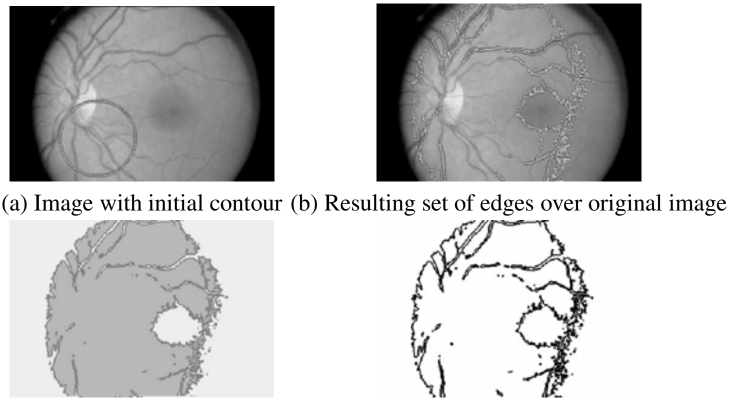
Fig. 6. Segmentation of a human brain MR image using the modified Mumford-Shah functional implementation



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

Fig. 7. Segmentation of a human eye vessel image using the standard Mumford-Shah functional implementation



(a) Image with initial contour (b) Resulting set of edges over original image

(c) Resulting set of regions (d) Resulting set of edges

Fig. 8. Segmentation of a human eye vessel image using the modified Mumford-Shah functional implementation

Table 1. Specifications for all the tests on the different imagery. In all the experiments and for both algorithms (standard and modified) the initial level set was long, and threshold $T = 0.001$ for all cases.

Image	No. Iterations	μ	Fractal Dimension	D_t
Mandelbrot	2000	0.01	1.4836	2
Julia	3000	0.001	1.4791	2
BrainMRI	1000	0.01	1.6871	0.5
Eye vessel	1000	0.000000001	1.3452	0.5

5 Discussion

A modification of the Mumford-Shah functional that includes Fractal Analysis to improve the segmentation of images with fractal or semi-fractal objects has been presented in this paper. We have described how the fractal dimension is calculated and incorporated in the functional minimization to drive the algorithm to use both, changes in the image intensities and the fractal characteristics of the objects.

Experimental results confirm that the proposed modification improves the quality of the segmentation in images with fractal objects (Mandelbrot and Julia sets) or semi fractal such as medical images (human brain MR and eye vessel images). For the true fractal images, we can clearly notice the segmentation enhancement, but for the case of the medical imagery is different. In the brain MR image we can observe that the resulting segmentation from the standard Mumford-Shah algorithm looses the different shades of gray, therefore it returns a less accurate result when we try to segment the white matter. Our modified algorithm was successful enough to segment accurately the white matter, but it retains some of the skull as well. The case of the human eye vessel image is a more complex problem due to the variability with artifacts and low contrast images, and despite that the modified algorithm is not able to segment all the different artifacts of the image, it presents a considerable improvement against the standard algorithm, that is not good enough again to catch the different gray levels hence loosing almost all the components of the image. It is relevant to note that this modified version of the algorithm might not perform well in images that do not contain fractal or semi-fractal objects, even returning less accurate results than the standard algorithm for such images.

In [21] evaluations of multiple contour detection and image segmentation algorithms on the Berkeley Segmentation Dataset(BSDS300) using a precision-recall framework are presented. This benchmark operates by comparing machine generated contours to human ground-truth data and allows evaluation of segmentations in the same framework by regarding region boundaries as contours. But the test presented in this work do not include images with fractal or semi-fractal objects such as medical images.

Hence, future working directions include the utilization of the aforementioned benchmark to compare human ground-truth from medical images and our method; or the usage of the concept of *lacunarity*, another fractal property that roughly speaking indicates how well does a fractal fills the space can improve the segmentation quality. Finally, another possible direction of improvement is to consider a different measurement than averaging the colour levels as the standard functional does.

References

1. Song, B., Chan, T.: A fast algorithm for level set based optimization. UCLA CAM Report, pp. 02–68 (2003)
2. Chan, T.F., Esedoglu, S.: A multiscale algorithm for muford-shah image segmentation, Mathematics Department, UCLA, pp. 3–77 (2003)
3. An, J.-h., Chen, Y.: Region Based Image Segmentation Using a Modified Mumford-Shah Algorithm. In: Sgallari, F., Murli, A., Paragios, N. (eds.) SSVM 2007. LNCS, vol. 4485, pp. 733–742. Springer, Heidelberg (2007)
4. Hongmei, Z., Mingxi, W.: Improved mumford-shah functional for coupled edge-preserving regularization and image segmentation. The Key Laboratory of Biomedical Information Engineering, Ministry of Education, Xi'an, China (2006)
5. Yongsheng Pan, J.D.B., Djouadi, S.M.: Bottom-up hierarchical image segmentation using region competition and the mumford-shah functional, vol. 2, pp. 117–121. IEEE Computer Society (2006)
6. Delrieux, C.: Image segmentation through automatic fractal dimension classification. In: In Argentine Symposium on Computing Technology, Buenos Aires, 32 JAIIO, Jornadas Argentinas de Informatica e Investigacion Operativa (2003)
7. Ferrell, R.K., Gleason, S.S., Tobin, K.W.: Application of fractal encoding techniques for image segmentation. In: Proceedings of the SPIE IEEE/SME/SPIE 6th International Conference on Quality Control by Artificial Vision (2003)
8. Vuduc, R.: Image segmentation using fractal dimension. In: GEOL 634, Cornell University (1997)
9. Mandelbrot, B.B.: The fractal geometry of nature. Freeman, Oxford (1982)
10. Bennett, S.H., Eldridge, M.W., Puente, C.E., Riedi, R.H., Nelson, T.R., Beotzman, B.W., Milstein, J.M., Singhal, S.S., Horsfield, K., Woldenberg, M.J.: Origin of Fractal Branching Complexity in the Lung (2001) (preprint)

11. Kiselev, V.G., Hahn, K.R., Auer, D.P.: Is the brain cortex a fractal? NeuroImage 20(3), 1765–1774 (2003)
12. Losa, G.A., Nonnenmacher, T.F.: Fractals in biology and medicine. In: 4th of Mathematics and Biosciences in Interaction, Birkhäuser (2005)
13. West, B.J.: Fractal Physiology and Chaos in Medicine. World Scientific, Singapore (1990)
14. Zhang, L., Liu, J.Z., Dean, D., Sahgal, V., Yue, G.H.: A three-dimensional fractal analysis method for quantifying white matter structure in human brain. Journal of Neuroscience Methods 150(2), 242–253 (2006)
15. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. Springer, Heidelberg (2006)
16. Giorgi, E.D., Ambrosio, L.: Un nuovo tipo di funzionale del calcolo delle variazioni. Atti. Accad. Naz, Lincei Rend. Cl Sci. Fis. Mat. Natur. 82, 199–210 (1988)
17. Ambrosio, L.: Variational problems in SBV and image segmentation. Acta Applicandae Mathematicae 17, 1–40 (1989)
18. Feder, J.: Fractals. Plenum Press, New York (1988)
19. Falconer, K.: Fractal Geometry: Mathematical Foundations and Applications, 2nd edn. John Wiley & Sons, Chichester (1990)
20. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Processing 10, 266–277 (2001)
21. Arbelá Andez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5), 898–916 (2011)

Robust RML Estimator – Fuzzy C-Means Clustering Algorithms for Noisy Image Segmentation

Dante Mújica-Vargas, Francisco Javier Gallegos-Funes,
Alberto J. Rosales-Silva, and Rene Cruz-Santiago

Mechanical and Electrical Engineering Higher School
National Polytechnic Institute of Mexico
Av. IPN s/n, Edificio Z, acceso 3, 3$^{\underline{er}}$ piso; SEPI-Electronica,
Col. Lindavista, 07738, México D.F. México
adorinam@yahoo.com.mx, fcogf@hotmail.com

Abstract. Image segmentation is a key step for many images analysis applications. So far, there does not exist a general method to segment suitable all images, regardless if these are corrupted or noise free. In this paper, we propose to modify the Fuzzy C-means clustering algorithm and the FCM_S1 variant by using the RML-estimator. The idea to our method is to get robust clustering algorithms able to segment images with different type and levels of noises. The performance of the proposed algorithms is tested on synthetic and real images. Experimental results show that the proposed algorithms are more robust to the noise presence and more effective than the comparative algorithms.

Keywords: robust estimators, RML-estimator, Fuzzy C-Means, segmentation, noise.

1 Introduction

Image segmentation is the process of segmenting an image into several disjoint regions whose characteristics such as intensity, color, texture, etc., are significantly different with respect to the same characteristics. It is a key step in early vision problem and it has been widely investigated in the field of image processing [1]. Numerous segmentation techniques have been developed and reported in the literature. But, there does not exist a general algorithm that can excellently perform the segmentation task for all type of images.

Fuzzy clustering as a soft segmentation method has been widely studied and successfully applied to image segmentation. Among the fuzzy clustering methods, Fuzzy C-Means (FCM) [2] algorithm is the most popular method because it is simple, easy to program, and can retain much more information than hard methods. Although fuzzy clustering methods work well on the most of noise-free images, they have a serious limitation: they do not incorporate any information about spatial context, which cause them sensitivity to the noise or outliers data. Then it is necessary to modify the objective function to incorporate local information of the image to get better results.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 474–486, 2011.

In this paper, we propose to modify the fuzzy c-means clustering algorithm, and the FCM_S1 [3], [4] using a RML estimator [5], [6], [7], [8]. The objective is to get robust algorithms to segment images under different noise conditions.

The experimental results show that the proposed algorithms are more effectives and robust to the noise than all reference algorithms.

2 Proposed Method

2.1 RML-Estimator

How it was proposed in [7], [9], the Median M-type (MM) – estimator can enhance the properties of the M- and R-estimators,

$$\theta_{medM} = med\{X_i\tilde{\psi}(X_i - med\{\vec{X}\}), i = 1, \ldots, N\} \tag{1}$$

where, X_i is the data sample, $\vec{X}$ is the data vector, and $\tilde{\psi}$ is the influence function. The robustness of the L-filter is improved by means of the RM-estimate (1). The representation of the L-filter is,

$$\theta_L = \sum_{i=1}^{N} a_i X_{(i)} \text{ with } a_i = \int_{i-1/N}^{i/N} h(\lambda)d\lambda \Big/ \int_0^1 h(\lambda)d\lambda \tag{2}$$

where, $X_{(i)}$, *i=1,…,n* is the i-th order statistics of the data sample (ascending order), $a_i, i = 1, \ldots, N$ are the weighted coefficients of filter, and $h(\lambda)$ is the noise probability distribution function [0,1]→R satisfying $\int_0^1 h(\lambda)d\lambda \neq 0$.

To introduce the RM-estimator [9] in the scheme of L-filter, the ordered data sample of L-filter should be presented as function of an influence function [9].

$$\theta_L = \sum_{i=1}^{N} a_i \psi(X_i) X_i \tag{3}$$
$$\psi(u) = \begin{cases} c & |u| \leq r \\ 0 & otherwise \end{cases}$$

where $N = (2L + 1)^2$ is the filtering window size, $\psi(X_i)X_i$ is the ordered data sample, $\psi(u)$ is the influence function, *c* is a constant, and *r* is connected with the range of $\psi(u)$. Then, the RM L-filter can be obtained by merging the L-filter (3) and the RM-estimator (1),

$$\theta_{RML} = \frac{med\{a_i[X_i\psi(X_i - med\{\vec{X}\})]\}}{a_{med}} \tag{4}$$

where, $X_i\psi(X_i - med\{\vec{X}\})$ are the selected pixels in accordance with the influence function in the sliding filter window, the coefficients a_i are computed using the Laplacian, Uniform and Exponential distribution functions in $h(\lambda)$, and a_{med} is the median of coefficients a_i used as scale constant. Table 1 shows the influence function used in this paper.

Table 1. Influence function used in the RML-estimator

Influence function	Formulae
Hampel's three part redescending	$\psi_{\alpha,\beta(r)}(x) = \begin{cases} x & 0 \le \|x\| < \alpha \\ \alpha\, sgn(x) & \alpha \le \|x\| < \beta \\ \alpha \dfrac{r - \|x\|}{r - \beta} sgn(x) & \beta \le \|x\| < r \\ 0 & r \le \|x\| \end{cases}$

To improve the properties of impulsive noise suppression of the RML-estimator an impulse detector is used.

$$[(rank(x_{ij}) \le s) \vee (rank(x_{ij}) \ge N - s)] \wedge (|x_{ij} - med\{x_n\}| \ge U_2) \tag{5}$$

where, x_{ij} is the central pixel in the filtering window, $s > 0$ y $U_2 \ge 0$ are thresholds, N is the length of the data sample and $med\{x_n\}$ is the median of pixels into the filtering window.

2.1 Robust Fuzzy Clustering Algorithms

Fuzzy C-Means is a method for data classification, where each data belongs to a cluster to some degree, which is specified by a membership value. This algorithm performs the iteration of two indispensable conditions to minimize the following objective function.

$$\begin{gathered} J_f(U,V,X) = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^{m} \left\| x_i - v_j \right\|^2 \\ subject\ to \sum_{i=1}^{N} u_{ij} > 0\ \forall_j\ and \sum_{j=1}^{c} u_{ij} = 1\ \ \forall_i \end{gathered} \tag{6}$$

where, $X = \{x_i | i = 1, \dots, N\}$ denotes the set of N feature vectors, c is the number of classes, $m \in [1, \infty)$ is a weighting exponent called the fuzzifier, $\left\| x_i - v_j \right\|^2$ is the square of the Euclidean distance from feature vector x_i to the center of the class v_j and $V = (v_1, \dots, v_j)$ is a vector with all center classes. $U = [u_{ij}]$ is a Nxc matrix denoting the constrained fuzzy c-partition. The value of u_{ij} denotes the degree of membership of x_i to the class v_j. To cover the noise sensitivity of the FCM algorithm, the RML-estimator is applied on the feature vector (intensity pixel) x_i. Based on the standard fuzzy c-means algorithm (6) and the RML-estimator (4). The objective function of the new algorithm can be written as:

$$J_{rf}(U,V,\Theta)=\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|\theta_{RML\,i}-v_j\right\|^2 \tag{7}$$

$$subject\ to\sum_{i=1}^{N}u_{ij}>0\ \forall_j\ \ and\ \sum_{j=1}^{c}u_{ij}=1\ \ \forall_i$$

where, $\Theta=\{\theta_{RML\,i}|i=1,\dots,N\}$ is the RML-estimator applied on the intensity feature vector. Taking in account both constrains, the membership matrix and the cluster prototypes can be calculated with the following equations.

$$v_j=\frac{\sum_{i=1}^{N}u_{ij}^{m}\ \theta_{RML\,i}}{\sum_{i=1}^{N}u_{ij}^{m}} \tag{8}$$

$$u_{ij}=\frac{\left\|\theta_{RML\,i}-v_j\right\|^{-\frac{2}{m-1}}}{\sum_{l=1}^{c}\left\|\theta_{RML\,i}-v_l\right\|^{-\frac{2}{m-1}}} \tag{9}$$

A shortcoming of the FCM_S1 algorithm [3], [4] proposed by Chen and Zhang (10), is that the effect of the noisy image could be more than the mean-filtered image and the median-filtered image getting then an unsuitable segmentation.

$$J(U,V,X)=\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|x_i-v_j\right\|^2+\alpha_1\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|\bar{x}_i-v_j\right\|^2+\alpha_2\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|\tilde{x}_i-v_j\right\|^2$$

$$subject\ to\sum_{i=1}^{N}u_{ij}>0\ \forall_j\ and\sum_{j=1}^{c}u_{ij}=1\ \ \forall_i \tag{10}$$

where, $\bar{x}_i$ and $\tilde{x}_i$ are the mean and median of the neighboring pixels lying with a window around x_i, respectively. The parameters α_1 and α_2 control the effect of the mean and median of the neighboring pixels. So, our propose is to modify the objective function of this algorithm changing the x_i, $\bar{x}_i$ y $\tilde{x}_i$ terms by $\theta_{RML}-Uniform$, $\theta_{RML}-Exponential$, and $\theta_{RML}-Laplacian$ terms to get a more robust segmentation.

$$\begin{aligned}J_{rdf}(U,V,\Theta)=&\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|\theta_{RML-U\,i}-v_j\right\|^2+\alpha_1\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|\theta_{RML-E\,i}-v_j\right\|^2\\&+\alpha_2\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{m}\left\|\theta_{RML-L\,i}-v_j\right\|^2\\&subject\ to\sum_{i=1}^{N}u_{ij}>0\ \forall_j\ \ and\ \sum_{j=1}^{c}u_{ij}=1\ \ \forall_i\end{aligned} \tag{11}$$

where, θ_{RML-U}, θ_{RML-E} and θ_{RML-L} are the θ_{RML} estimators (4) computed using the Uniform, Exponential and Laplacian distributions respectively, α_1 and α_2 are parameters that control the effect of , θ_{RML-E} and θ_{RML-L} terms. By minimizing (11), the membership matrix and the cluster prototypes can be stated as.

$$v_j = \frac{\sum_{i=1}^{N} u_{ij}^{m}\ [\theta_{RML-U\,i} + \alpha_1 \theta_{RML-E\,i} + \alpha_2 \theta_{RML-L\,i}]}{(1+\alpha_1+\alpha_2)\sum_{i=1}^{N} u_{ij}^{m}} \tag{12}$$

$$u_{ij} = \frac{\left[\left\|\theta_{RML-U\,i} - v_j\right\|^2 + \alpha_1\left\|\theta_{RML-E\,i} - v_j\right\|^2 + \alpha_2\left\|\theta_{RML-L\,i} - v_j\right\|^2\right]^{-\frac{1}{m-1}}}{\sum_{l=1}^{c}[\|\theta_{RML-U\,i} - v_l\|^2 + \alpha_1\|\theta_{RML-E\,i} - v_l\|^2 + \alpha_2\|\theta_{RML-L\,i} - v_l\|^2]^{-\frac{1}{m-1}}} \tag{13}$$

3 Experimental Results

The performance of the proposed method was tested on synthetic and real images. In both cases the quantitative results were compared using the optimal segmentation accuracy (SA). SA was measured as the total number of correctly classified pixels divided by the total number of pixels (see Ahmed et al., 2002) [4].

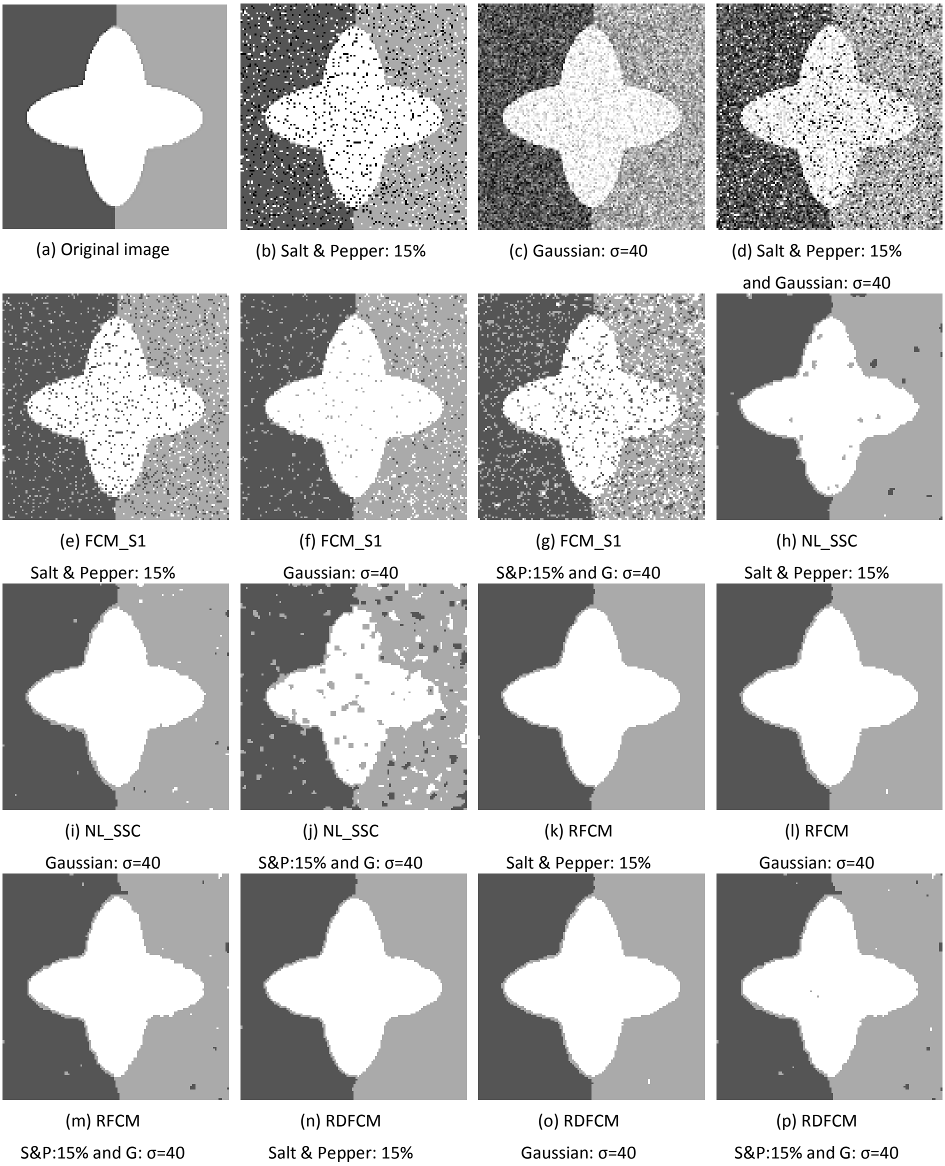
$$SA = \frac{Total\,number\,of\,correctly\,classified\,pixels}{total\,numer\,of\,pixels}\ x100\% \tag{14}$$

3.1 Evaluation on a Synthetic Image

A synthetic image (8-bit pixels) containing 128x128 pixels is depicted in Fig.1(a). This one has *three clusters* with gray values of 85, 170, and 255 respectively. The original image was corrupted by different levels of Gaussian noise, Salt & Pepper noise and a mix of these ones. The results are compared with the FCM_S1, FCM_S2, Nyström, NCut, and NL_SSC algorithms [3], [4], [10]. For both proposed algorithms (7) and (11) the parameters used were stated as: c=3, m=2, ε=1e-6 in the clustering (initialized randomly); r=5, α=0.16r, β=0.8r in the RML-estimation, s=4 and U_2=5 in the impulse detection, α_1=0.02 and α_2=0.05 in the RDFCM algorithm (11). Table 2 shows the SAs values for all algorithms and Fig.2 depicts the visual results in the case of mixed noise.

Table 2. SA % on the synthetic image

Algorithm	Gaussian			Salt & Pepper			Mixed		
	σ=10	σ=20	σ=40	5%	10%	15%	σ=10 + 5%	σ=20 + 10%	σ=40 + 15%
FCM_S1	97.95	97.91	93.59	96.39	95.13	92.81	96.33	95.10	90.30
FCM_S2	97.98	97.95	93.48	96.39	95.13	92.83	96.34	95.25	91.30
Nyström	76.72	74.98	71.68	96.82	94.56	92.13	76.87	75.62	72.93
NCut	99.69	99.62	98.84	94.81	94.90	94.11	95.58	94.31	92.84
NL_SSC	99.80	99.71	99.47	98.42	98.01	97.26	98.71	98.09	97.37
RFCM	**99.95**	**99.85**	**99.66**	**99.96**	**99.92**	**99.89**	**99.87**	**99.83**	**99.49**
RDFCM	**99.93**	**99.85**	**99.61**	**99.97**	**99.92**	**99.90**	**99.85**	**99.80**	**98.46**



(a) Original image

(b) Salt & Pepper: 15%

(c) Gaussian: σ=40

(d) Salt & Pepper: 15% and Gaussian: σ=40

(e) FCM_S1 Salt & Pepper: 15%

(f) FCM_S1 Gaussian: σ=40

(g) FCM_S1 S&P:15% and G: σ=40

(h) NL_SSC Salt & Pepper: 15%

(i) NL_SSC Gaussian: σ=40

(j) NL_SSC S&P:15% and G: σ=40

(k) RFCM Salt & Pepper: 15%

(l) RFCM Gaussian: σ=40

(m) RFCM S&P:15% and G: σ=40

(n) RDFCM Salt & Pepper: 15%

(o) RDFCM Gaussian: σ=40

(p) RDFCM S&P:15% and G: σ=40

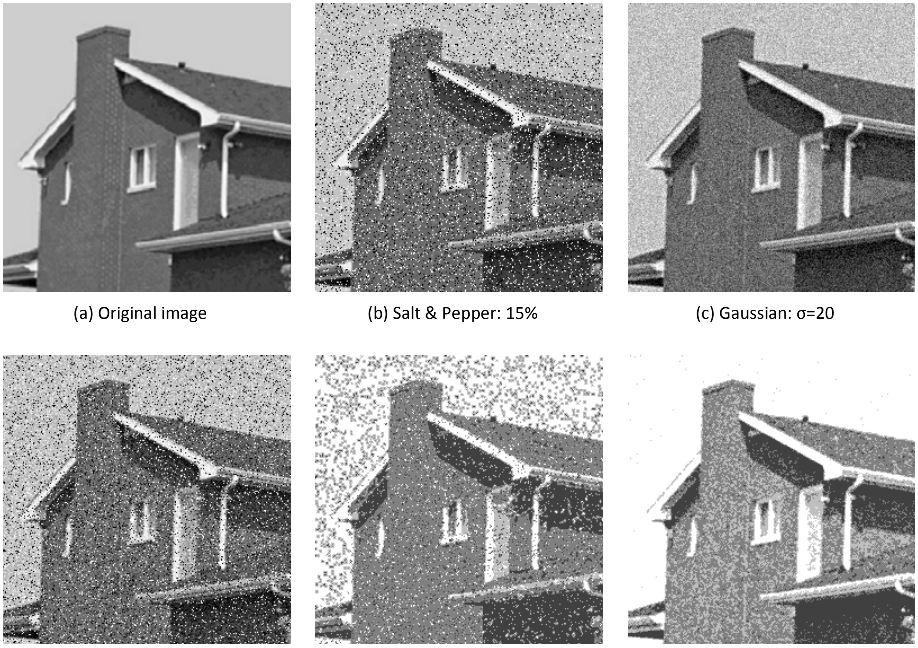
Fig. 1. Segmentation results in a synthetic image

3.2 Evaluation on a Real Image

The robustness on a real image (Fig.2(a)) with a format of 8-bit pixels was tested using the *home* real image with 256x256 pixels size corrupted by different levels of Gaussian noise, Salt & Pepper noise and a mix of these ones. This image was tested with *four clusters*. The results are compared with FCM_S1, FCM_S2, Nyström, NCut, and NL_SSC algorithms [3], [4], [10], [11]. For proposed algorithms the parameters used were stated as: c=4, m=2, ε=1e-6 in the clustering (initialized randomly); r=5, α=0.16r, β=0.8r in the RML-estimation, s=4 and U_2=5 in the impulse detection, α_1=0.8 and α_2=0.4 in the RDFCM algorithm. Table 3 shows the calculated SAs for all algorithms and Fig. 2 depicts the visual results.

Table 3. SA % on the real image

Algorithm	Gaussian			Salt & Pepper			Mixed		
	σ=5	σ=10	σ=20	5%	10%	15%	σ=5 + 5%	σ=10 + 10%	σ=20 + 15%
FCM_S1	97.51	97.56	70.98	96.72	82.22	71.17	96.63	77.15	66.11
FCM_S2	97.52	97.50	70.61	96.78	78.17	70.35	97.14	69.87	62.74
Nyström	80.27	76.39	70.28	93.66	91.76	88.43	77.28	73.69	68.75
NCut	97.44	95.83	92.48	91.36	88.90	84.33	92.65	89.95	86.75
NL_SSC	98.85	97.62	96.17	99.28	98.64	98.06	98.97	97.80	96.82
RFCM	**99.83**	**99.76**	**98.72**	**99.92**	**99.90**	**99.81**	**99.88**	**99.75**	**98.60**
RDFCM	**99.77**	**99.72**	**98.91**	**99.61**	**99.55**	**99.62**	**99.84**	**99.78**	**98.96**



(a) Original image

(b) Salt & Pepper: 15%

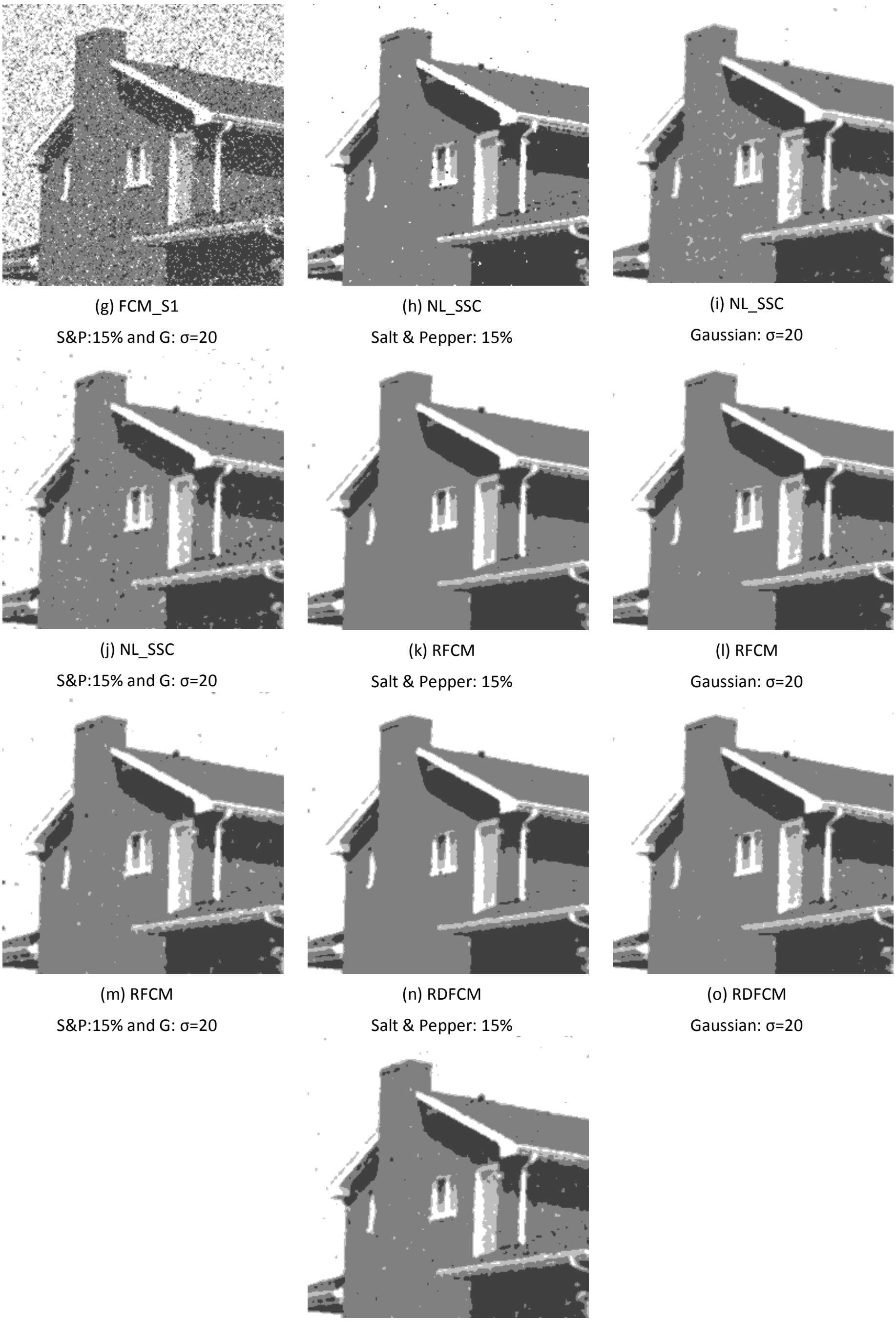
(c) Gaussian: σ=20

(d) Salt & Pepper: 15% and Gaussian: σ=20

(e) FCM_S1 Salt & Pepper: 15%

(f) FCM_S1 Gaussian: σ=20

Fig. 2. Segmentation results in a real image



(g) FCM_S1
S&P:15% and G: σ=20

(h) NL_SSC
Salt & Pepper: 15%

(i) NL_SSC
Gaussian: σ=20

(j) NL_SSC
S&P:15% and G: σ=20

(k) RFCM
Salt & Pepper: 15%

(l) RFCM
Gaussian: σ=20

(m) RFCM
S&P:15% and G: σ=20

(n) RDFCM
Salt & Pepper: 15%

(o) RDFCM
Gaussian: σ=20

(m) RDFCM
S&P:15% and G: σ=20

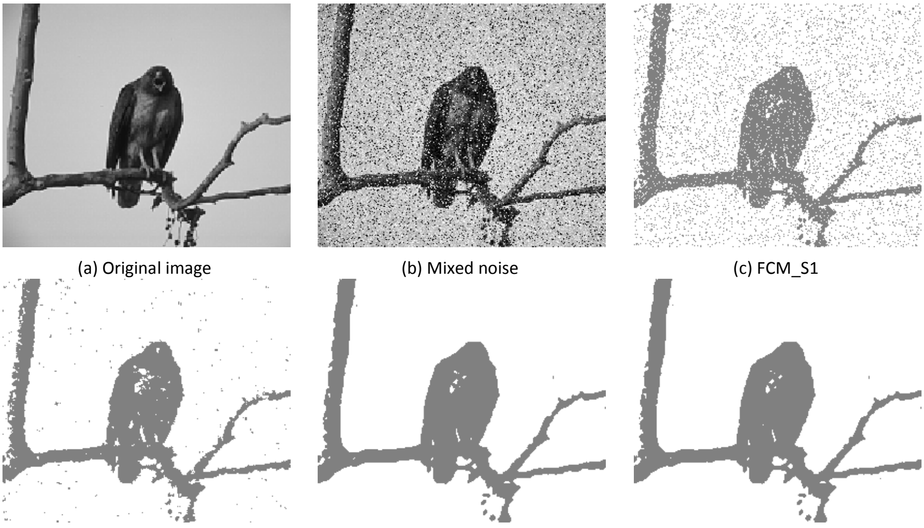
Fig. 2. (*continued*)

3.3 Evaluation with Berkeley Image Segmentation Data Set 500 (BSDS500)

In this experiment, we evaluate the image segmentation performance of the proposed algorithms and the results are compared with the FCM_S1 and NL_SSC algorithms, using a subset of the Berkeley image segmentation dataset. The images were corrupted by Salt & Pepper, Gaussian and a mix of them. For proposed algorithms the parameters used were stated as: m=2, ε=1e-6, c=3 for all images (except 42049-image, where c=2) in the clustering (initialized randomly); r=5, α=0.16r, β=0.8r in the RML-estimation, s=4 and U_2=5 in the impulse detection, α_1=0.8 and α_2=0.4 in the RDFCM algorithm. Table 4 shows the calculated SAs and Figs. 3, 4, 5, 6, and 7 depict the visual results.

Table 4. SA % on the real images

Algorithm	Noise	Image				
		42049	35010	24063	8068	302003
FCM_S1	Salt & Pepper: 15%	92.76	97.92	86.29	92.51	95.94
	Gaussian: σ=20	97.95	98.07	98.97	97.76	98.17
	Mixed: 15% + : σ=20	90.77	96.49	84.97	89.99	93.06
NL_SSC	Salt & Pepper: 15%	99.05	98.04	98.94	98.38	97.84
	Gaussian: σ=20	98.02	98.75	98.15	98.13	97.29
	Mixed: 15% + : σ=20	98.21	97.82	97.96	98.04	96.33
RFCM	Salt & Pepper: 15%	**99.95**	**99.40**	**99.97**	**99.95**	**99.52**
	Gaussian: σ=20	**99.92**	**99.48**	**99.77**	**99.42**	**97.98**
	Mixed: 15% + : σ=20	**98.30**	**98.75**	**99.05**	**99.58**	**97.48**
RDFCM	Salt & Pepper: 15%	**99.68**	**98.58**	**99.65**	**99.60**	**98.23**
	Gaussian: σ=20	**99.91**	**98.39**	**99.78**	**99.53**	**97.86**
	Mixed: 15% + : σ=20	**98.34**	**98.11**	**99.17**	**99.50**	**97.20**


(a) Original image

(b) Mixed noise

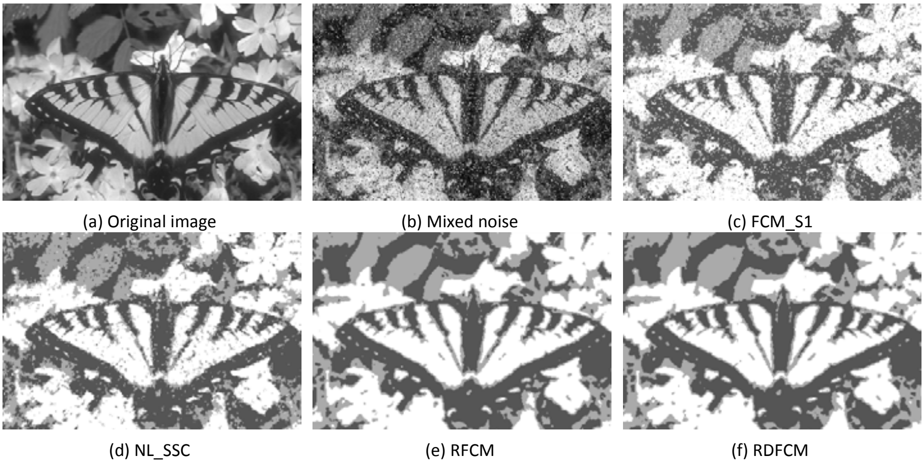
(c) FCM_S1

(d) NL_SSC

(e) RFCM

(f) RDFCM

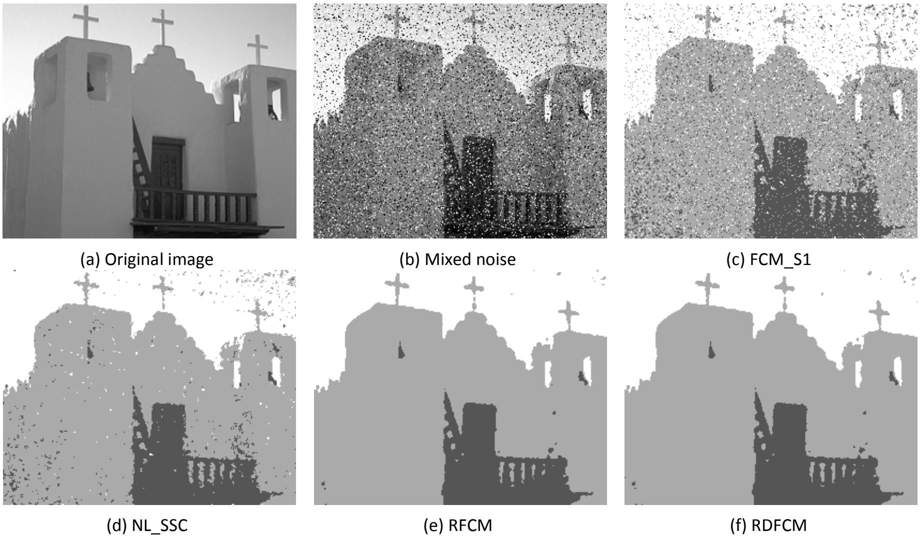
Fig. 3. Image 42049 with $c=2$



(a) Original image (b) Mixed noise (c) FCM_S1

(d) NL_SSC (e) RFCM (f) RDFCM

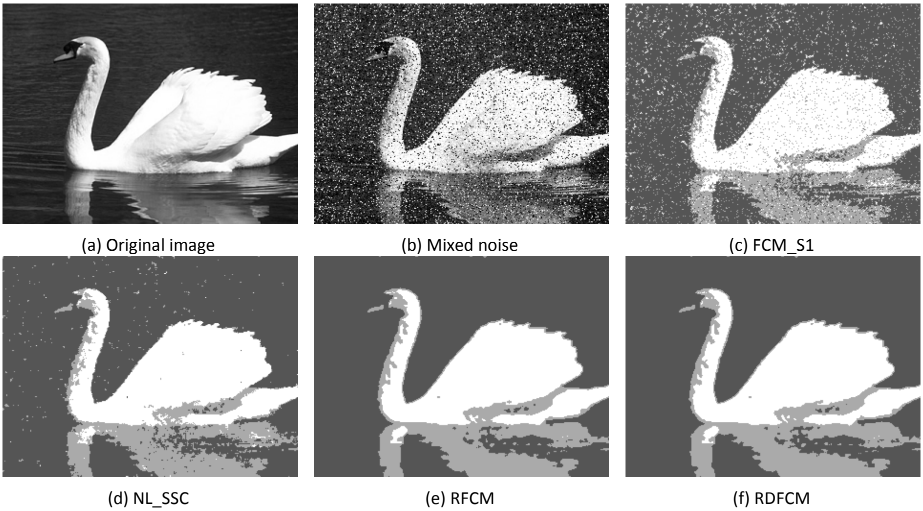
Fig. 4. Image 35010 with *c=3*



(a) Original image (b) Mixed noise (c) FCM_S1

(d) NL_SSC (e) RFCM (f) RDFCM

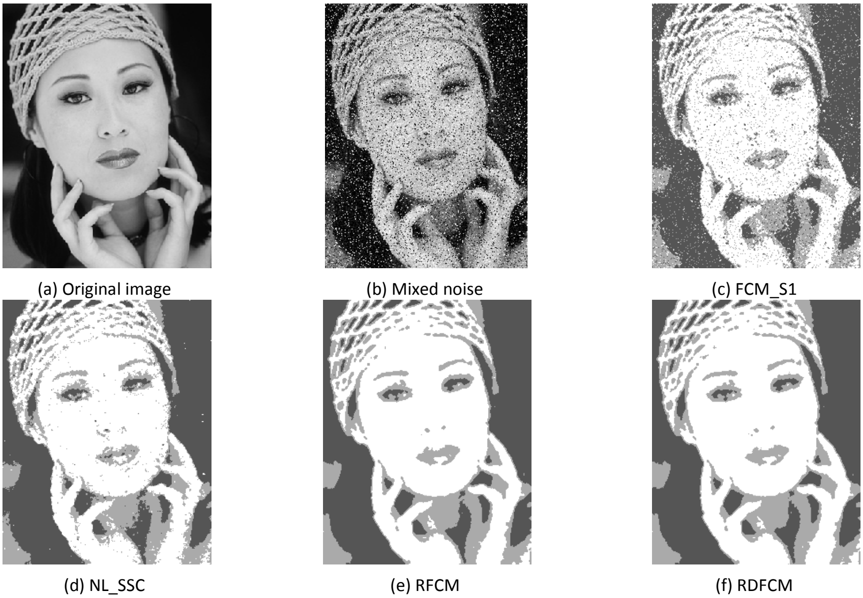
Fig. 5. Image 24063 with *c=3*



(a) Original image (b) Mixed noise (c) FCM_S1

(d) NL_SSC (e) RFCM (f) RDFCM

Fig. 6. Image 8068 with *c=3*



(a) Original image (b) Mixed noise (c) FCM_S1

(d) NL_SSC (e) RFCM (f) RDFCM

Fig. 7. Image 302003 with *c=3*

4 Discussion of Results

The experimental results on both test images shown that the proposed algorithms obtained better results than the comparative algorithms. Also, was observed that the incorporation of the RML-estimator into de FCM_S1 algorithm gave it greater ability so segment noisy images in comparison of its original version. The SA% values obtained by our algorithms in both experiments is close to 100%, so, this is appreciated in better conservation of the details in the images after segmentation.

As a future work another indexes will be evaluate to measure the segmentation quality, some of these are the Probabilistic Rank Index (PRI), Variation of Information (VOI), Global Consistency Error (GCE) and the Boundary Displacement Error (BDE) [12].

5 Conclusions

This paper presented a method for incorporating the RML-estimator into the Fuzzy C-means and its variant FCM_S1, making the more robust to segment noisy images. The performance of the proposed methods was better than the reference algorithms. However, to give them greater ability to segment color images as well as other types and levels of noise should be modified and optimized.

Acknowledgments. This work is supported by National Polytechnic Institute of Mexico and Conacyt.

References

1. Yang, F., Jiang, T.: Pixon-based Image Segmentation with Markov random fields. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China (2002)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
3. Wang, Z., Lu, R.: A New Algorithm for Image Segmentation Based on Fast Fuzzy C-Means Clustering. School of Electronics and Information Engineering, Southwest University, Chongqing, China (2008)
4. Cai, W.L., Chen, S.C., Zhang, D.Q.: Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. Department of Computer Science & Engineering. Nanjing University of Aeronautics & Astronautics, Nanjing, China (2007)
5. Gallegos-Funes, F., Varela-Benitez, J., Ponomaryov, V.: Rank M-Type L (RM L)-Filter for Image Denoising. IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences E91-A(12), 3817–3819 (2008)
6. Gallegos-Funes, F.J., Ponomaryov, V.: Real-time image filtering scheme based on robust estimators in presence de noise impulsive. Real Time Imaging 8(2), 78–90 (2004)
7. Gallegos-Funes, F.J., Varela-Benitez, J.L., Ponomaryov, V.: Real-time image processing based on robust linear combinations of order statistics. In: Proc. SPIE Real-Time Image Processing 2006, vol. 6063, pp. 177–187 (2006)
8. Varela-Benitez, J.L., Gallegos-Funes, F.J., Ponomaryov, V.: RML-filters for real rime imaging. In: Proc. IEEE 15th International Conference on Computing, CIC 2006, pp. 43–48 (2006)

9. Gallegos-Funes, F., Ponomaryov, V., De-La-Rosa, J.: ABST M-type K-nearest neighbor (ABSTM-KNN) for image denoising. IEICE Trans. Funds. Electronics Comms. Computer Science E88A(3), 798–799 (2005)
10. Liu, H.Q., Jiao, L.C., Zhao, F.: Non-local spatial spectral clustering for image segmentation. Neurocomputing 74, 461–471 (2010)
11. Liu, H.Q., Jiao, L.C., Zhao, F.: A novel fuzzy clustering with non-local adaptive spatial constraint for image segmentation. Signal Processing 91, 988–999 (2011)
12. Le Capitaine, H., Frélicot, C.: A fast fuzzy c-means algorithm for color image segmentation. Laboratoire Mathémathiques, Image et Applications, Université de La Rochelle, France (2011)

Processing and Classification of Multichannel Remote Sensing Data

Vladimir Lukin[1], Nikolay Ponomarenko[2], Andrey Kurekin[3], and Oleksiy Pogrebnyak[4]

[1,2] National Aerospace University, Dept of Transmitters, Receivers and Signal Processing,
17 Chkalova St, 61070 Kharkov, Ukraine
lukin@xai.kharkov.ua, nikolay@ponomarenko.info
[3] Plymouth Marine Laboratory, Prospect Place, The Hoe,
Plymouth United Kingdom PL1 3DH
akurekin@gmail.com
[4] Instituto Politecnico Nacional, Centro de Investigacion en Computacion,
Ave. Juan de Dios Batiz S/N, C.P. 07738, Mexico, D.F., Mexico
olek@pollux.cic.ipn.mx

Abstract. Several main practical tasks, important for effective pre-processing of multichannel remote sensing (RS) images, are considered in order to reliably retrieve useful information from them and to provide availability of data to potential users. First, possible strategies of data processing are discussed. It is shown that one problem is to use more adequate models to describe the noise present in real images. Another problem is automation of all or, at least, several stages of data processing, like determination of noise type and its statistical characteristics, noise filtering and image compression before applying classification at the final stage. Second, some approaches that are effective and are able to perform well enough within automatic or semi-automatic frameworks for multichannel images are described and analyzed. The applicability of the proposed methods is demonstrated for particular examples of real RS data classification.

Keywords: RS data classification, image compression, noise filtering.

1 Introduction

Remote sensing data obtained by airborne and spaceborne systems are widely used nowadays [1-4]. Potential customers of RS data are governmental boards, local authorities, space agencies, nature protection organizations, marine traffic services, meteorologists, etc. Most of them require more reliable information and its more operative offering [2]. Information can be provided to customers in different forms: as raw or pre-filtered images, compressed data, classification maps [2, 3], etc. A most convenient form depends upon an application RS data are used for, time available for solving a final task, qualification and resources of customer's personnel and so on. Thus, an owner or provider of RS data has to be able to carry out various kinds of data processing.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 487–498, 2011.

Usefulness of RS data and effectiveness of solving final tasks depend upon several factors. Among them, we can mention the following ones: a) information content of original (raw) data which is determined by RS operation mode, range of wavelengths covered by an imaging system, number of its channels, spatial resolution, b) noise statistical characteristics of acquired images; adequateness of noise models, availability and correctness of a priori information about model parameters; c) effectiveness of methods applied for RS data processing. By processing we mean a wide set of possible operations. Depending upon application, this set might include evaluation of noise characteristics, pre-filtering, compression, registration, geo-referencing, calibration, segmentation, etc. [4].

A general way to increase information content of RS data is to use multichannel RS complexes. By the term "multichannel" we mean that a set of images of a sensed terrain is formed either simultaneously or sequentially. Different RS imagers used in optical, infrared, and radio wave bands [3, 4] perform nowadays in multichannel mode where the number of channels (bands) is from few to hundreds.

On one hand, multichannel RS data make possible to solve a wider set of practical tasks. This is done by information retrieval from all or the most informative subsets of images. On the other hand, volume of RS data radically grows if channel number increases. Then, it becomes more difficult to manipulate multichannel RS data: to transfer them, to select proper subsets of channels (components, sub-bands), to choose and implement appropriate methods of image enhancement and processing.

Multichannel data processing is usually performed in several stages [4] where operations (stages) are distributed (divided) between on-board and on-land processing units. While dividing these functions between on-board and on-land processors, one has to keep in mind possible limitations and a final goal RS data are intended for. Clearly, on-board facilities are more restrictive due to limited power and weight as well as necessity to carry out processing in fully automatic mode [5].

The main operations that have impact on the final task, e.g., image classification are RS data compression (if lossy) and filtering. Below we focus on lossy compression since even the most powerful techniques of lossless coding are nowadays unable to provide a compression ratio (CR) larger than 3.5…4 [6] and this is often not enough for practical applications due to downlink channel limitations.

It has been shown recently that in the case of quite noisy original RS data both lossy compression and filtering can result in better classification accuracy compared to classification of original RS data [7-9]. In particular, it has been recently understood [2] that it is worth using near-lossy and lossy compression of multichannel RS data under conditions that the introduced losses relate to removal of noise and they do not lead to distorting valuable information. The authors [8] have run into an effect that lossy compression of hyperspectral data under certain conditions can lead to positive outcome of increased probability of correct classification compared to classification of original data. However, the authors [8] have not given explanation of this fact. The presented paper partly answers this question. Another open question is how to perform such lossy compression to simultaneously satisfy abovementioned requirements, especially if this should be done in automatic manner. It is also desirable to provide high enough CR and appropriate accuracy of solving the final tasks of RS like multichannel image classification, anomaly or target detection, etc.

Image pre-filtering can also lead to increasing probability of correct classification [9]. However, questions are how large can be this increase in cases of filtering and lossy compression, is it worth using them separately or jointly, what are possible strategies in the sense of distributing these operations between on-board and on-land processing units. Thus, below we concentrate on considering possible strategies and stages of multichannel RS data processing. Since hyperspectral data processing is a complicated case, we analyze a simpler example of three-channel Landsat TM data. Moreover, we assume that all component images are corrupted by additive independent identically distributed (i.i.d.) noise.

2 Possible Strategies of On-Board/On-Land Processing

There are, at least, three possible strategies for on-board/on-land processing of multichannel RS data [10, 11].

A first strategy (**strategy 1**) [10] presumes that a multichannel image is a subject to lossy compression without any pre- and post-filtering. In Section 3, we focus on this strategy more in detail. This strategy assumes that the first stage is blind evaluation of noise variance. It is applied separately to each component (sub-band) image and can be carried out sequentially or in parallel. As a result, a set of estimates of noise standard deviations (SDs) $\hat{\sigma}(n), n=1,...,N$ is obtained where N denotes a number of components of multichannel image. Then, the component (sub-band) images can be either grouped or compressed separately [10, 12]. More details concerning grouping are given in [10, 12]. In any case, grouping and consequent 3D compression provide about 1.5…2.5 increase of compression ratio (CR) compared to component-wise lossy compression (benefit is larger for larger N). However, 3D compression is more complex. In any case, quantization step (QS) is set depending upon estimated standard deviation of noise. More details on setting QS will be provided later. But it is worth noting that within this strategy QS is selected so that coder operates in the neighbourhood of an optimal operation point, i.e. providing as efficient noise removal as it is possible for lossy compression [13]. If grouping is not used, each component image is compressed with 2D coder setting QS equal to $QS(n)=C_1\hat{\sigma}(n), n=1,...,N$ where C_1 is a parameter of about 4.5.

An advantage of this strategy is its relative simplicity. No on-board filtering is carried out (recall that on-board filtering requires additional resources). Another advantage is that this strategy provides quite large CR. For example, it ensures CR from 4.5 to 9 for component-wise compression and from 8 to 25 for compression with adaptive sub-band grouping for 224-channel AVIRIS data (http://aviris.jpl.nasa.gov/). However, there is also one serious drawback. Decompressed image quality cannot be practically improved by post-filtering applied on-land.

According to an alternative strategy (**strategy 2**), multichannel data are compressed in near-lossless manner. The goal in this case is to provide favourable conditions for on-land post-filtering of decompressed data (this is the first difference compared to strategy 1). Again, the first stage is blind evaluation of $\hat{\sigma}(n), n=1,...,N$. Then, the data can be either compressed component-wise or grouped. The difference compared to the **strategy 1** is that $QS(n)=C_2\hat{\sigma}(n), n=1,...,N$ (for component-wise compression) where the parameter C_2 is considerably smaller than C_1 for the **strategy 1**. The recommended value is $C_2 \approx 1.3$.

One drawback is that the **strategy 2** provides sufficiently smaller CRs than the **strategy 1**, namely, from 3.3 to 7.2 for component-wise compression and from 5.4 to 14.2 for compression with grouping (these values have been obtained for standard AVIRIS data (conventional test images Moffett Field, Cuprite Mine, Lunar Lake, Jasper Ridge). An obvious advantage is that the **strategy 2** offers good potential for consequent effective filtering of decompressed images on-land (if this is needed). Moreover, filtering of decompressed multichannel data on-land can be more efficient than on-board because of several reasons. First, time and resources that can be spent on denoising on-land are not as limited as on-board. Thus, more sophisticated methods can be applied [14]. Second, filtering can be carried out in interactive manner with more careful setting of filter parameters.

For a third strategy (**strategy 3**), pre-filtering is to be carried out on-board. Then pre-processed data are compressed in a lossy manner and passed downlink. On-land the received data can be either decompressed or disseminated in compressed form.

As for the strategies 1 and 2, the first stage of processing is again blind evaluation of $\hat{\sigma}(n), n = 1,...,N$. This operation is mainly used for carrying out component-wise or 3D filtering of images. Note that most advanced methods for image filtering [14-17] require a priori knowledge or pre-estimation of noise standard deviation. Besides, the estimates $\hat{\sigma}(n), n = 1,...,N$ are used to set a coder quantization step. In particular, for component-wise compression $QS(n) = C_3 \hat{\sigma}(n), n = 1,...,N$ where C_3 is a parameter. It is recommended to set it approximately equal to 1.5 [12]. Then, distortions introduced by lossy compression of pre-filtered images are practically negligible and they have small impact on further classification.

The **strategy 3** has been tested for AVIRIS images as well. It provides CR from 3.4 to 10.0 for component-wise compression and from 5.6 to 20.0 for 3D compression with sub-band adaptive grouping.

One advantage of the **strategy 3** is its ability to produce higher quality of images than the **strategy 1**. In turn, a drawback of the **strategy 3** is that it requires considerable resources and time for on-board data processing (compared to two other strategies) since pre-filtering of multichannel images is to be performed on-board.

In general, **strategies 2** and **3** provide approximately the same quality of pre-processed (compressed and filtered) data. Attained CR values are about the same as well. Thus, below we can compare classification accuracy for compressed and then filtered multichannel data (**strategy 2**) keeping in mind that for the **strategies 2** and **3** compression introduces negligible distortions into data.

3 Coders Used and Filtering Effect for the Strategy 1

It is clear that a used coder influences performance of any multichannel data classification technique. Below we consider two advanced coders, both based on discrete cosine transform (DCT).

A first coder is called AGU [18]. It has both 2D [18] and 3D [10] versions. DCT is applied in 32x32 pixel blocks, more advanced probability models and image deblocking after decompression are used to further improve its performance (rate-distortion characteristic).

Another considered coder is ADCT [19]. It implies partition scheme and adaptive size of blocks starting from 4x4 and up to 64x64.

One reason for using these DCT-based coders is that CR for them is varied by QS. This is convenient for the considered strategies since recommended QSs are proportional to obtained estimates of noise standard deviations.

As it has been first demonstrated in [20] and later confirmed by other researchers (see, e.g., [13, 21]), lossy compression of noisy images possesses filtering effect. Due to this, lossy compression is to be characterized by peak signal-to-noise ratio (PSNR) calculated with respect to noise-free image ($PSNR_{nf}$) rather than conventional PSNR. $PSNR_{nf}$ can be calculated only for test images when the corresponding noise-free image $I_{ij}^{nf}, i = 1.,...,I_{im}, j = 1,...,J_{im}$ is available

$$PSNR_{nf} = 10\log_{10}((I^{\max} - I^{\min})^2 / MSE_{dec\,nf}), \tag{1}$$

$$MSE_{dec\,nf} = \sum_{i=1}^{I_{im}} \sum_{j=1}^{J_{im}} (I_{ij}^{dec} - I_{ij}^{nf})^2 / (I_{im} J_{im} - 1), \tag{2}$$

where $I^{\max}, I^{\min}$ denote the maximal and minimal values of the images, respectively, I_{ij}^{dec} is the *ij*-th pixel of decompressed image, $I_{im} J_{im}$ define the image size. Al-Chaykh et al [20] have also shown that for different types of noise there exists a so-called optimal operation point (OOP). By OOP such CR (or bit rate expressed in bits per pixel (bpp)) is meant for which $PSNR_{nf}$ attains maximum or, equivalently, $MSE_{dec\,nf}$ reaches minimum. Moreover, it has been shown recently that OOP might also exist for other than PSNR metrics; in particular, those ones that describe visual quality of lossy compressed noisy images [22]. The only peculiarity is that for different metrics OOP (CR or bpp) can be slightly different. For example, OOP according to visual quality metrics corresponds to smaller CR (larger bpp and smaller QS) than OOP for $PSNR_{nf}$ [22].

One practical problem was earlier how to reach this OOP when noise-free image is not available. However, this problem has been recently solved, at least, for coders controlled by QS [13, 21, 22]. For pure additive noise case, this can be done by setting $C_1 \approx 4.5$ if one needs to get into the neighbourhood of OOP according to $\mathrm{PSNR}_{\mathrm{nf}}$ [13]. C_1 should be slightly smaller (about 3.5) if OOP for visual quality metrics is to be provided. Moreover, multi-stage procedures involving homomorphic transforms have been proposed for pure multiplicative and Poisson noise [21].

As it has been mentioned above, lossy compression of noisy images leads to image filtering. However, this filtering due to lossy compression is not as efficient as conventional filtering. The best way to understand the reasons is to compare DCT based filtering [17] to lossy compression [18]. First, the best way of DCT based filtering is to apply fully overlapping blocks [17]. Meanwhile, blocks are not overlapped in the case of lossy compression. Second, in case of hard thresholding, quantization (assigning zero values) is applied only to relatively small DCT coefficients in the case of filtering. If compression is applied, quantization of all DCT coefficients is carried out. Due to these differences, $PSNR_{nf}$ in the best case of filtering (if threshold is set as $T=\beta\sigma(n)$, β=2.6 and full overlapping of blocks is used) is about 3 dB larger than $PSNR_{nf}$ for OOP in case of lossy compression. Since noise suppression is less efficient for lossy compression, one can expect that RS data filtering (strategies 2 and 3) is able to produce better pre-conditions for RS data classification. Let us see is this really so.

4 Classification of Compressed and Filtered Multichannel Images

It is clear that maximal $PSNR_{nf}$ due to compression or filtering does not necessarily result in solving the final tasks of RS data processing as, e.g., classification, object and anomaly detection [23], etc., in the best way. To study how conventional metrics that characterize efficiency of image filtering and compression are interconnected with quantitative measures that describe classification accuracy we have carried out special experiments. The test image has been formed using three channels (sub-bands) of Landsat TM image. These component images relate to central wavelengths 0.66 μm, 0.56 μm, and 0.49 μm (optical bands) and they have been associated with R, G, and B of color images represented in Fig. 1,a.

All components have been artificially corrupted by pure i.i.d. additive noise (noise is independent in component images). Five classes have been defined, namely, bare soil (class 1), grass (class 2), water surface (class 3), roads and urban areas (class 4), and bushes (class 5). Support vector machine (SVM) classifier and radial basis

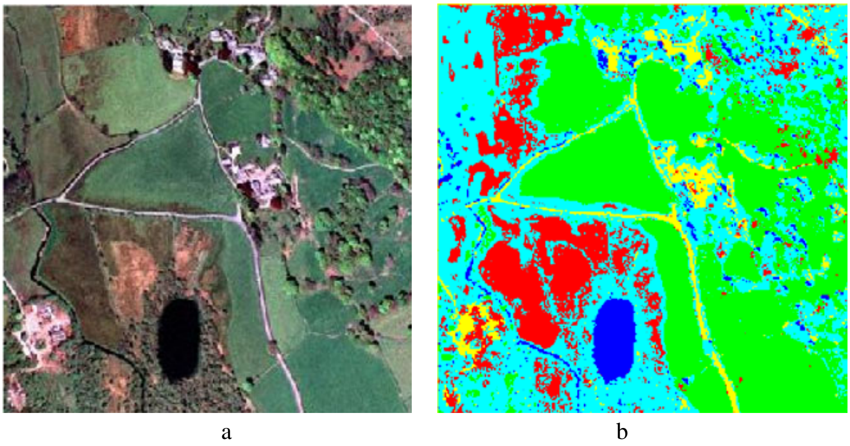


Fig. 1. The test three-channel image in RGB representation (a) and the corresponding classification map for noise-free multichannel data (b)

function (RBF) neural network (NN) classifiers [24] have been applied. These types of classifiers are widely used in image classification and are known to be efficient enough. Training has been performed for noise-free image. Training set contained about 10 times smaller number of samples (vector of pixel intensities) than the validation set. Other details of their training and parameter selection can be found in [7, 9]. Classification has been carried out pixel-wise using three values of component image intensities in a given pixel (voxel) as feature vector.

The influence of lossy compression on classification accuracy has been examined for different compression ratios and two considered coders (AGU and ADCT) applied component-wise. Recall that for both of them CR is determined by QS.

Let us present data obtained for additive noise variance equal to 100. The simulation results are presented in Tables 1 and 2. Here, P_c denotes probability of correct classification in aggregate (for all considered classes).

Table 1. Compression and classification characteristics for different QS (coder AGU)

QS	P_c (NN)	P_c (SVM)	*bpp*	$PSNR_{nf}$, dB
Noisy image	0.766	0.729	-	28.10
25	0.777	0.742	1.63	28.22
35	0.855	0.819	1.06	29.03
45 (OOP)	0.871	0.835	0.75	29.11
55	0.873	0.836	0.58	28.87

Table 2. Compression and classification characteristics for different QS (coder ADCT)

QS	P_c (NN)	P_c (SVM)	*bpp*	$PSNR_{nf}$, dB
25	0.791	0.761	1.57	28.27
35	0.870	0.833	1.02	29.48
45 (OOP)	0.887	0.852	072	29.65
55	0.890	0.854	0.55	29.02

As it is seen, OOP according to $PSNR_{nf}$ for both coders is observed for $QS = 45 = 4.5\sigma$. Maxima of P_c and $PSNR_{nf}$ practically coincide. Thus, by providing the maximal $PSNR_{nf}$ due to proper setting of QS one reaches the neighbourhood of P_c maximum. Joint analysis of data in Tables 1 and 2 shows that larger $PSNR_{nf}$ usually leads to larger P_c. The coder ADCT produces by about 4% better CR and slightly larger P_c for the same QS. However, ADCT is more complex.

Although aggregate P_c possesses maximum for $QS \approx 5\sigma$, dependences of probabilities for different classes behave in a specific manner. They are given in Table 3 for the coder ADCT, RBF NN classifier and noise variance equal to 100. Analysis of data shows the following. First, probabilities for different classes vary a lot. Whilst water surface is classified almost perfectly, there are considerable probabilities of misclassifications between other classes. For some classes as Bare Soil and Grass which are quite homogeneous the corresponding probabilities (P_1 and

P_2) increase if larger QS is used and, respectively, noise reduction due to lossy compression improves. On the contrary, for quite heterogeneous classes (Roads and Buildings, Bushes) the corresponding probabilities either remain practically the same (P_5) or even slowly decrease (P_4) due to distortions introduced by lossy compression.

Table 3. Compression and classification characteristics for different classes (coder ADCT)

QS	P_1	P_2	P_3	P_4	P_5
25	0.68	0.83	>0.99	0.72	0.76
35	0.76	0.91	>0.99	0.72	0.77
45 (OOP)	0.82	0.97	>0.99	0.71	0.77
55	0.83	0.97	>0.99	0.71	0.77

It is also interesting to see the obtained classification maps and to compare them. Fig. 2 shows the classification map obtained for original noisy image (a) and the multichannel image compressed component-wise by the ADCT coder with QS=45, i.e. set in quasioptimal manner (b). In both cases, the RBF NN classifier has been applied. Obviously, there are quite many pixel-wise misclassifications for the map in Fig. 2,a. Their percentage has been considerably (by about two times) decreased due to filtering effect of lossy compression (see Fig. 2,b).

Consider the classification results for the **strategies 2** and **3** that employ filtering. The obtained data are collected in Table 4. Two filters are considered: the component-wise DCT based filter with hard thresholding [17] and the 3D DCT filter [26]. It is seen that the use of the 3D DCT-based filter is preferable irrespectively to the classifier applied. Comparison of P_c for the same noise variance and classifier clearly shows that classification results for the strategies 2 and 3 are better than for **strategy 1** (compare data in Tables 4 and Tables 1 and 2 (for OOP).

Table 4. P_c for strategies 2 and 3 for NN and SVM classifiers

Filtering method	Noise variance 49		Noise variance 100	
	P_c (NN)	P_c (SVM)	P_c (NN)	P_c (SVM)
Component-wise	0.90	0.89	0.90	0.88
3-D	0.91	0.91	0.91	0.91

Fig. 3 represents classification maps for filtered images processed by the 3D DCT-based filter [26]. Although the classification results depend upon classifier used, there are considerably less misclassifications compared to the map in Fig. 2,a. Mainly misclassifications are observed for the classes Bare Soil and Bushes.

Note that P_c for the filtered images is almost the same as for noise-free classified data. This shows that filtering is very efficient.

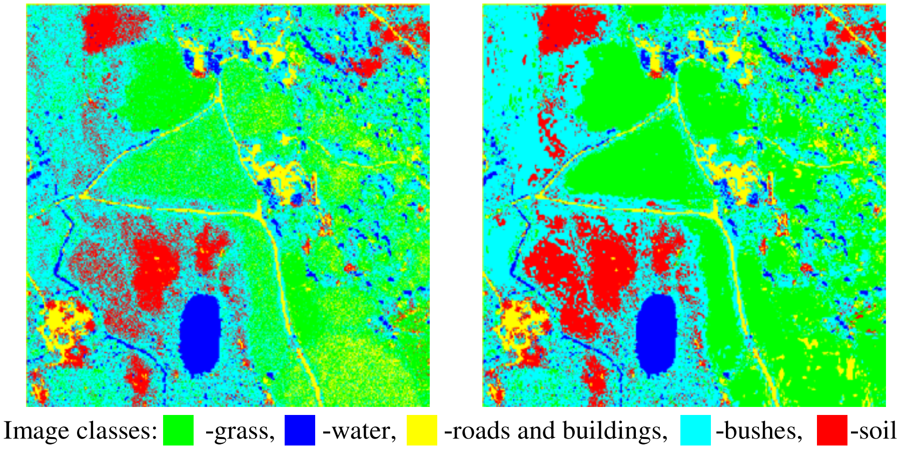
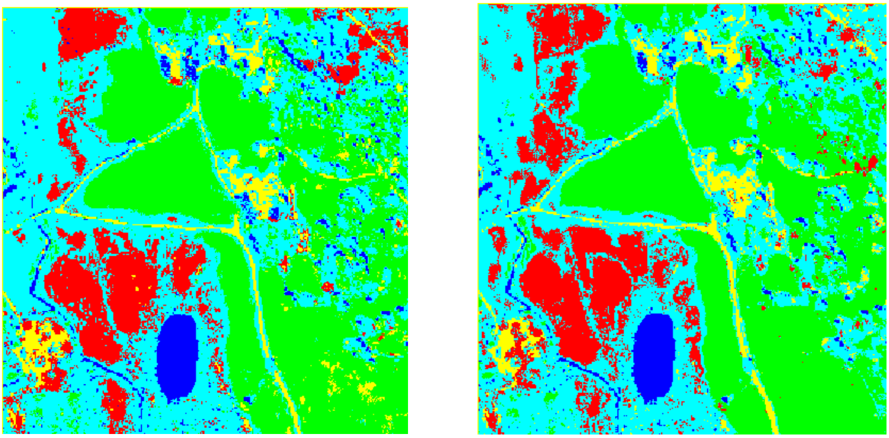


Image classes: 🟩 -grass, 🟦 -water, 🟨 -roads and buildings, 🟦 -bushes, 🟥 -soil

(a) (b)

Fig. 2. The classification maps (a) for the noisy image and (b) the optimally compressed image



(a) (b)

Fig. 3. Classification maps for noisy image classified by RBF NN (a) and SVM (b) after processing by the 3D DCT filter

5 Discussion and Conclusions

In this paper, we have considered the simple case of pixel-wise classification of three-channel image corrupted by moderate intensity noise and pre-processed (lossy compressed and/or filtered). The presented example and analysis results show that both lossy compression (under condition of properly set parameters) and filtering can lead to positive effect of classification accuracy increasing due to noise reduction.

In practice, there are several circumstances that can cause problems and require more careful dealing with lossy compression and filtering of multichannel images.

First, noise can be not pure additive, pure multiplicative or Poisson. Quite complicated dependences of noise local variance on local mean have been observed for real life hyperspectral data [27, 28]. Then one needs more accurate blind estimators of noise characteristics [28] and more careful approaches to lossy compression and filtering [28, 29]. Different homomorphic (variance stabilizing) transforms can be useful to simplify the situation.

Second, noise intensity (and statistical characteristics) in component images of hyperspectral data vary in wide limits [1, 3, 28]. There are component (sub-band) images for which noise intensity is large enough (it is more correct to talk about PSNR low enough). Then the results presented above are helpful for setting coder parameters. It is worth applying filtering for the corresponding sub-band images. Concerning practice, our recommendation is to apply pre- or post-filtering within strategies 2 and 3 if original PSNR (calculated with taking into account dynamic range) for a sub-band image is less than 35 dB. Otherwise, it is not worth applying any filter and the main task of compression is not noise reduction but restricting of introduced distortions level with providing as large CR as possible.

Above we have considered three different strategies for automatic processing and compression of multichannel RS images. All of them have certain advantages and drawback. Choice of a proper strategy depends upon many factors as priority of requirements, restrictions imposed on on-board and on-land processing, final tasks to be solved by exploiting multichannel RS data passed downlink and disseminated between customers. A positive feature of the considered methods of lossy compression and filtering is that all of them are based on DCT which can be easily implemented in software and hardware. In future, we plan to analyze multichannel data with more sub-bands and a larger number of images.

Acknowledgements. This work was supported by Instituto Politécnico Nacional as a part of the research project SIP#20113659.

References

1. Aiazzi, B., Baronti, S., Lastri, C., Santurri, L., Alparone, L.: Low complexity lossless/near-lossless compression of hyperspectral imagery through classified linear spectral prediction. In: Proceedings of IGARSS, p. 4 (2005)
2. Christophe, E.: Hyperspectral Data Compression Tradeoff. In: Prasad, S., et al. (eds.) Optical Remote Sensing. Advances in Signal Processing and Exploitation Techniques Series: Augmented Vision and Reality, vol. 3, pp. 9–30. Springer, Heidelberg (2011)
3. Chang, C.-I. (ed.): Hyperspectral Data Exploitation: Theory and Applications. Wiley-Interscience (2007)
4. Kulemin, G.P., Zelensky, A.A., Astola, J.T., Lukin, V.V., Egiazarian, K.O., Kurekin, A.A., Ponomarenko, N.N., Abramov, S.K., Tsymbal, O.V., Goroshko, Y.A., Tarnavsky, Y.V.: Methods and Algorithms for Pre-processing and Classification of Multichannel Radar Remote Sensing Images, TTY Monistamo, Tampere, Finland. TICSP Series, vol. 28, p. 116 (2004)
5. Lukin, V.V., Abramov, S.K., Ponomarenko, N.N., Uss, M.L., Zriakhov, M., Vozel, B., Chehdi, K., Astola, J.T.: Methods and automatic procedures for processing images based on blind evaluation of noise type and characteristics. SPIE Journal of Applied Remote Sensing 5, 53502 (2011)

6. Mielikäinen, J.: Lossless compression of hyperspectral images using lookup tables. IEEE Signal Processing Letters 13, 157–160 (2006)
7. Lukin, V., Ponomarenko, N., Kurekin, A., Lever, K., Pogrebnyak, O., Fernandez, L.P.S.: Approaches to Classification of Multichannel Images. In: Martínez-Trinidad, J.F., Carrasco Ochoa, J.A., Kittler, J. (eds.) CIARP 2006. LNCS, vol. 4225, pp. 794–803. Springer, Heidelberg (2006)
8. García-Vílchez, F., Muñoz-Marí, J., Zortea, M., Blanes, I., González-Ruiz, V., Camps-Valls, G., Plaza, A., Serra-Sagristà, J.: On the Impact of Lossy Compression on Hyperspectral Image Classification and Unmixing. IEEE Geoscience and Remote Sensing Letters 8(2), 253–257 (2011)
9. Fevralev, D.V., Lukin, V.V., Ponomarenko, N.N., Vozel, B., Chehdi, K., Kurekin, A., Shark, L.: Classification of filtered multichannel images. In: Bruzzone, L. (ed.) Proc. of SPIE, Image and Signal Processing for Remote Sensing XVI, vol. 7830, p. 78300M (2010)
10. Ponomarenko, N., Lukin, V., Zriakhov, M., Kaarna, A., Astola, J.: An automatic approach to lossy compression of AVIRIS images. In: Proceedings of IGARSS, Spain, pp. 472–475 (2007)
11. Ponomarenko, N., Lukin, V., Zriakhov, M., Kaarna, A., Astola, J.: Automatic approaches to on-board/on-land lossy compression of AVIRIS images. In: Proceedings of IGARSS, Boston, USA, p. 4 (2008)
12. Ponomarenko, N., Zriakhov, M., Lukin, V., Kaarna, A.: Improved Grouping and Noise Cancellation for Automatic Lossy Compression of AVIRIS Images. In: Blanc-Talon, J., Bone, D., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2010, Part II. LNCS, vol. 6475, pp. 261–271. Springer, Heidelberg (2010)
13. Ponomarenko, N., Lukin, V., Zriakhov, M., Egiazarian, K., Astola, J.: Estimation of accesible quality in noisy image compression. In: Proceedings of EUSIPCO, Florence, Italy, p. 4 (2006)
14. Foi, A.: Pointwise Shape-Adaptive DCT Image Filtering and Signal-Dependent Noise Estimation, Thesis for the degree of Doctor of Technology, Tampere University of Technology, Tampere, Finland (2007), `http://dspace.cc.tut.fi/dpub/handle/123456789/115`
15. Sendur, L., Selesnick, I.W.: Bivariate shrinkage functions for wavelet based denoising exploiting interscale dependency. IEEE Transactions on Signal Processing 50(11), 2744–2756 (2002)
16. Oktem, R., Egiazarian, K., Lukin, V.V., Ponomarenko, N.N., Tsymbal, O.V.: Locally adaptive DCT filtering for signal-dependent noise removal. EURASIP Journal on Advances in Signal Processing 2007, 10 (2007)
17. Lukin, V.V., Oktem, R., Ponomarenko, N., Egiazarian, K.: Image filtering based on discrete cosine transform. Telecommunications and Radio Engineering 66(18), 1685–1701 (2007)
18. Ponomarenko, N., Lukin, V., Egiazarian, K., Astola, J.: DCT Based High Quality Image Compression. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 1177–1185. Springer, Heidelberg (2005)
19. Ponomarenko, N.N., Lukin, V.V., Egiazarian, K., Astola, J.: ADCTC: a new high quality DCT based coder for lossy image compression. In: CD ROM Proceedings of LNLA, p. 6 (2008)
20. Al-Chaykh, O.K., Mersereau, R.M.: Lossy compression of noisy images. IEEE Transactions on Image Processing 7(12), 1641–1652 (1998)

21. Lukin, V., Ponomarenko, N., Zriakhov, M., Zelensky, A., Egiazarian, K., Astola, J.: Quasi-optimal compression of noisy optical and radar images. In: Proceedings of SPIE Conf. Image and Signal Processing for Remote Sensing XII, Sweden, vol. 6365 (2006)
22. Ponomarenko, N., Krivenko, S., Lukin, V., Egiazarian, K.: Lossy Compression of Noisy Images Based on Visual Quality: A Comprehensive Study. EURASIP Journal on Advances in Signal Processing, 13 (2010)
23. Christophe, E., Leger, D., Mailhes, C.: Quality criteria benchmark for hyperspectral imagery. IEEE Transactions on Geoscience and Remote Sensing 43(9), 2103–2114 (2005)
24. Bose, N.K., Liang, P.: Neural network fundamentals with graphs, algorithms and applications. McGraw-Hill (1996)
25. Schölkopf, B., Burges, J.C., Smola, A.J.: Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
26. Ponomarenko, N.N., Lukin, V.V., Zelensky, A.A., Koivisto, P.T., Egiazarian, K.O.: 3D DCT Based Filtering of Color and Multichannel Images. Telecommunications and Radio Engineering 67, 1369–1392 (2008)
27. Barducci, A., Guzzi, D., Marcoinni, P., Pippi, I.: CHRIS-Proba performance evaluation: signal-to-noise ratio, instrument efficiency and data quality from acquisitions over San Rossore (Italy) test site. In: Proceedings of the 3rd ESA CHRIS/Proba Workshop, Italy, p. 11 (2005)
28. Uss, M., Vozel, B., Lukin, V., Chehdi, K.: Local Signal-Dependent Noise Variance Estimation from Hyperspectral Textural Images. IEEE Journal of Selected Topics in Signal Processing 5(2) (in print, 2011)
29. Lukin, V., Krivenko, S., Zriakhov, M., Ponomarenko, N., Abramov, S., Kaarna, A., Egiazarian, K.: Lossy compression of images corrupted by mixed Poisson and additive noise. In: Proceedings of LNLA, Helsinki, pp. 33–40 (2009)

Iris Image Evaluation for Non-cooperative Biometric Iris Recognition System

Juan M. Colores[1], Mireya García-Vázquez[1],
Alejandro Ramírez-Acosta[2], and Héctor Pérez-Meana[3]

[1] Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI-IPN)
Avenida del Parque 1310, Tijuana, B.C. México 22510
[2] MIRAL R&D, Palm Garden, Imperial Beach, USA 91932
[3] Sección de Graduados de Mecánica y Eléctrica (ESIME-IPN), DF., México
{colores,mgarciav}@citedi.mx, ramacos10@hotmail.com,
hmperezm@ipn.mx

Abstract. During video acquisition of an automatic non-cooperative biometric iris recognition system, not all the iris images obtained from the video sequence are suitable for recognition. Hence, it is important to acquire high quality iris images and quickly identify them in order to eliminate the poor quality ones (mostly defocused images) before the subsequent processing. In this paper, we present the results of a comparative analysis of four methods for iris image quality assessment to select clear images in the video sequence. The goal is to provide a solid analytic ground to underscore the strengths and weaknesses of the most widely implemented methods for iris image quality assessment. The methods are compared based on their robustness to different types of iris images and the computational effort they require. The experiments with the built database (100 videos from MBGC v2) demonstrate that the best performance scores are generated by the kernel proposed by Kang & Park. The FAR and FRR obtained are 1.6% and 2.3% respectively.

Keywords: Iris recognition, Kernel, Convolution, Defocus, Quality, Video, MBGC.

1 Introduction

Nowadays, the development of better image quality metrics is an active area of research. The image quality plays a crucial role in the matching system, particularly in automated biometric systems such as iris recognition which performance is based upon matching fine texture information in the annular region between the pupil and the sclera. Some studies report that using a high quality image affects recognition accuracy and can improve system performance [1]. Then, it is necessary to select suitable images with high quality from an input video sequence before the next recognition processing. Otherwise, it can have a negative impact on segmentation algorithms [2,3].

In a recognition system, not all captured iris images are clear and sharp enough for recognition. In fact, in a real capturing iris images system, the person to recognize

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 499–509, 2011.

usually moves his head in different ways gives rise to non-ideal images (with occlusion, off-angle, motion-blur and defocus) for recognition. Defocus blur and Motion blur are the major source of iris image quality degradation [4]. Defocus blur occurs when the focal point is outside the "*depth of field*" of the object to be captured. Depth of field is the region of a camera which can capture a well-focused image. It is affected by aperture size, the smaller the aperture size the greater the depth of field. Motion blur can result either from the relative motion of an object or relative motion of the camera during exposure time. A sample set of all these image problems are shown in figure 1. As it shown, choosing an appropriate image with quality seems a challenge.

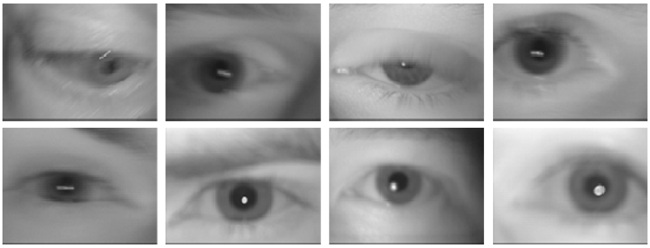


Fig. 1. Video sequences depicting various problems during capturing iris images

To address the problem of image quality, related work on this subject can be placed into two categories [3,5]: local and global analyses. Local methods try to classify each pixel of the iris providing additional information about each region of the iris texture. Zhu et al [6], propose a quantitative quality descriptor by analyzing the coefficients of particular areas of iris's texture by employing discrete wavelet decomposition. Chen et al [2,7], classify iris quality by measuring the energy of concentric iris bands obtained from two dimension wavelets. Ma et al [8], defined a quality descriptor according to characterize out-of-focus and motion blur and occlusions. Zhang and Salganicoff [9] examine the sharpness of the boundary between the pupil and the iris to determine defocus in images. Belcher and Du [10] propose a clarity measure by comparing the sharpness of iris image regions. The major feature of these approaches is that the evaluation of iris image quality is reduced to the estimation of a single or a pair of factors, such as out-of-focus blur, motion blur, and occlusion. Iris quality should not be limited to one or two quality factors. Moreover, the majority of previously methods require involvement of traditional segmentation methods that are iterative and thus computationally expensive.

The global methods are quick quality evaluation procedures for selecting the best images from a video sequence and eliminate very poor quality images. Global methods are mainly based on focus and motion blur estimation. Tenenbaum [11] proposes a method to determine the focus score using the gradient value of processed edge image. For checking the motion blur and defocus Jarvis [12], uses the Sum Modulus Difference. Nayar [13] adopts the Sum Modified Laplacian to measure the focus score; this method is based on the absolute values in the second derivative (Laplacian). However, such methods target on the entire iris image, and they can generate the wrong focus score in case of iris image. Especially, in case of users with glasses, if the lens is positioned for focusing the scratched glasses surface or the glasses image, such cases may make their focusing scores highest. To overcome this kind of problems, some methods have been proposed to exclusively check the focus score of iris image.

In this paper, we present the results of a comparative analysis of four global methods for iris image quality assessment to select clear images in the video sequence. These methods determine defocus and motion blur level in iris images. The goal is to provide a solid analytic ground to underscore the strengths and weaknesses of the most widely implemented methods for iris image quality assessment. The methods are compared based on their robustness to different types of iris images and the computational effort they require. These methods are based on convolution kernels where segmentation is not required because the operator is applied to the entire image, giving the possibility of its implementation at hardware level.

This paper is organized as follows. Section 2 explains the principles of kernel-based defocus measurements. Section 3 presents the main representative kernels used for of iris recognition systems. Methodologies for comparing kernels and results are given in Section 4, and Section 5 gives the conclusion.

2 Measurement of Blurring and Defocus in Iris Images

In [14], J. Daugman stated that an effective way to estimate the degree of focus in a broadband image is by measuring its total power in the 2D Fourier domain at higher spatial frequencies, because these are the most attenuated by defocus.

Figure 2 shows the total high frequency power in the 2D Fourier spectrum of two iris images assessing the focus of the image. This is a common method applied to image quality assessment [14-17]. In an iris recognition system if an iris image can pass a minimum focus criterion, it will be used for recognition. Thus, it needs a discrete formulation to obtain only the high frequency power.

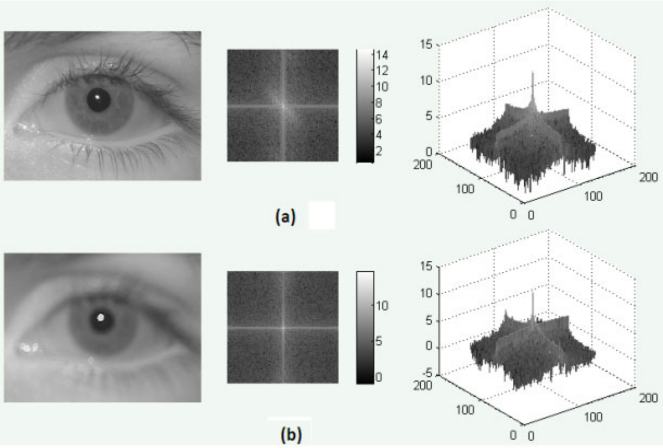


Fig. 2. a) Clear Image. b) Defocus Image. The total high frequency powers in the 2D Fourier spectrum are 8.772 x 10^6 and 4.509 x 10^6 respectively.

This can be solved filtering the low frequency part of the image, computing the total power in higher frequency bands of the processing image (low frequency filtered image) and setting a predefined threshold [14-17].

2.1 Filter Low Frequencies in Images

In image processing one of the most common linear operations is filtering. It is known that image filtering can be performed both in the frequency domain or spatial domain. Convolution in the spatial domain is a simple mathematical operation, in which a kernel (matrix) of numbers is multiplied by each pixel and its neighbors in a small region.

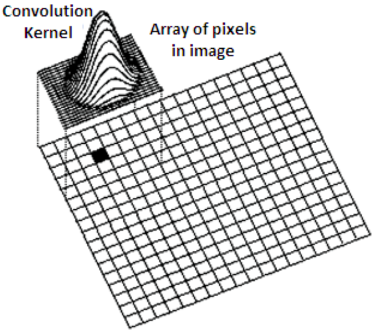


Fig. 3. Illustration of applying a convolution kernel to an image in the spatial domain

There are many spatial-domain kernels such as a high-pass filter that can be applied to an image by convolving the kernel with the original image. The figure 3, illustrates graphically the process for a single placement of the kernel.

The equation for the two-dimensional discrete convolution is given by:

$$C(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I(m,n)g(i-m,j-n) \tag{1}$$

where $I(m,n)$ is an image, $g(i,j)$ is the convolution kernel, and $c(i,j)$ is the filtered image.

2.2 Computing the Total Power of an Image

In mathematics, *Parseval's theorem* shows that the Fourier transform is unitary, i.e., the sum (or integral) of the square of a function equals the sum (or integral) of the square of its transform. The total power of the signal is equal to the total power of its Fourier transform along all its frequency components. The total power, P, in a 2D signal can be measured either in the spatial domain or the frequency domain. For an image $c(i,j)$ in the spatial domain with the MxN dimension, the total power is calculated using the *Parseval's theorem,* so the total power for 2D discrete space signals are calculated by the equation (2).

$$P = \frac{1}{M\,N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(i,j)\,|^2 \tag{2}$$

3 Convolution Kernels

In order to obtain the total power in higher frequency bands of the image, a proper high-pass convolution kernel is really important. In this section, we will give a brief description of the four convolution kernels most frequently presented in scientific literature to determine the defocus degree in eye-iris images.

3.1 Daugman's Convolution Kernel

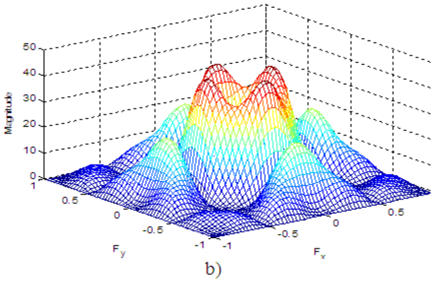
In his pioneering work [14], Daugman proved that the defocus primarily attenuates high spatial frequencies. In this band-pass filter, the central frequency is around 0.28125 with a bandwidth of 0.1875 in which the attenuation is less than 3db with respect to the central frequency. The $K(\mathrm{u}\,,\mathrm{v}\,)$, 2-D Fourier transform is represented as equation (3) and its spectrum response are shown in figure 4b.

$$K(u,v) = \frac{\sin(u)\sin(v)}{\pi^2 uv} - \frac{\sin(2u)\sin(2v)}{4\pi^2 uv} \tag{3}$$

Due to this relationship, he improved the convolution operation in real-time, id est., to reduce the computational complexity of the Fourier transform he proposed a high pass 8×8 convolution kernel to extract the high frequency of an image. The weights consist of two square box functions, one of size 8x8 with amplitude -1, and the other one of size 4x4 and amplitude of +4.

-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	3	3	3	3	-1	-1
-1	-1	3	3	3	3	-1	-1
-1	-1	3	3	3	3	-1	-1
-1	-1	3	3	3	3	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1

a)



b)

Fig. 4. a) The 8x8 Convolution kernel proposed by Daugman b) The frequency response (Fourier Spectrum)

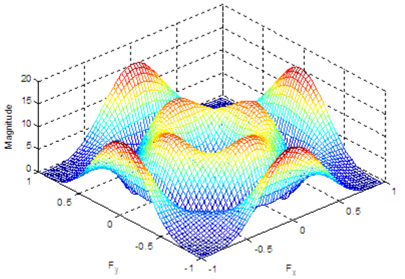
3.2 The Convolution Kernel of Wei et al.

Wei et al. [15] also suggest a convolution kernel, with a similar shape as Daugman's for detecting defocused on still and segmented images. Additionally, they detect other problems presented in iris images; motion blur and occlusion. Each problem has its own peculiarity, so, the three features are used to classify them using Support Vector Machine (SVM) that is a machine learning algorithm method used for classification [18]. The total quality of an image according with their method is a vector Q (q1, q2, q3), where the values represent the levels from defocus, motion blur and occlusion respectively.

To determine the defocus degree they proposed a 5×5 convolution kernel as shown in figure 5. Compared with Daugman's 8×8 convolution kernel is also a lower frequencies filter but computationally less demanding. The operator is formed by three box functions, one of size 5x5 with amplitude -1, one of size 3x3 with amplitude +3, and the last one of size 1x1 with amplitude -2.

-1	-1	-1	-1	-1
-1	2	2	2	-1
-1	2	0	2	-1
-1	2	2	2	-1
-1	-1	-1	-1	-1

a)



b)

Fig. 5. a) The 5x5 Convolution kernel proposed by Wei et al. b) The frequency response (Fourier Spectrum)

The kernel Fourier spectrum is shown in figure 5b. It is a band-pass filter with central frequency around 0.4375 and bandwidth of 0.3125 in which the attenuation is less than 3db with respect to the central frequency.

3.3 Laplacian of Gaussian Convolution Kernel

J. Wang et al. [16] propose a convolution kernel operator based on a Laplacian of Gaussian function (LoG). The Laplacian gives a 2-D isotropic measure of the second spatial derivative of an image. In the image the high intensities regions represent the rapid intensity change, which are used for edge detection. The Gaussian smooth filter is first applied to the image to reduce the noise sensibility of the second derivative. Laplacian $L(\mathrm{x}, \mathrm{y})$ of an image with pixel intensity values $I(\mathrm{x}, \mathrm{y})$ is given as follows.

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{4}$$

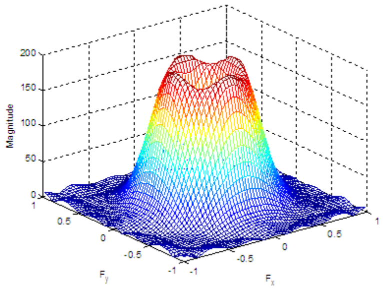
The combination of both filters gives a function centered on zero with standard deviation σ as is given by the equation 5:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

Since the image is represented as a set of pixels, the authors sought a discrete convolution kernel that can approximate the Laplacian operator. They set different values of the Gaussian. Finally they used σ = 1.4 that produce a LoG operator as shown in figure 6.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	2	4	5	5	5	4	2	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	2	4	5	5	5	4	2	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

a)



b)

Fig. 6. a) The 9x9 Convolution kernel based in Laplacian and Gaussian filter b) The frequency response (Fourier Spectrum)

3.4 The Convolution Kernel of Kang and Park

Kang & Park [17] propose 5x5 pixels sized convolution kernel as shown in figure 7. It is a band-pass filter and its central frequency is around 0.2144, with a bandwidth of 0.6076 in which the attenuation is less than 3 db with respect to the central frequency.

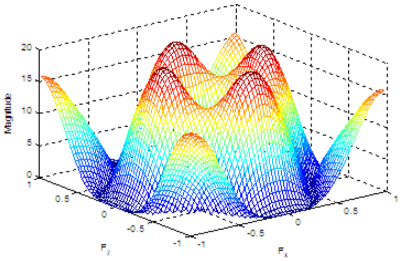
The $K(\mathrm{u},\mathrm{v}\,)$ in the 2-D Fourier domain is plotted en figure 7b and represented as equation (6).

$$K(u,v) = \frac{\sin\left(\frac{3}{2}u\right)\sin\left(\frac{3}{2}v\right)}{\frac{9}{4}\pi^2 uv} - \frac{\sin\left(\frac{5}{2}u\right)\sin\left(\frac{5}{2}v\right)}{\frac{25}{4}\pi^2 uv} - 4\cdot\frac{\sin\left(\frac{1}{2}u\right)\sin\left(\frac{1}{2}v\right)}{\frac{1}{4}\pi^2 uv} \tag{6}$$

The kernel consists of three square box functions, one of size 5x5 with amplitude -1, one of size 3x3 and amplitude +5, and other of size 1x1 and amplitude -5 (see figure 7). They argue that their 5x5 pixels convolution kernel contains more high frequency bands than the 8x8 pixels convolution kernel proposed by Daugman [14]. According to the authors the operator can detect much better the high frequency of iris texture, using less processing time due to the short sized kernel.

-1	-1	-1	-1	-1
-1	-1	4	-1	-1
-1	4	4	4	-1
-1	-1	4	-1	-1
-1	-1	-1	-1	-1

a)



b)

Fig. 7. a) The 5x5 Convolution kernel proposed by Kang & Park b) The frequency response (Fourier Spectrum)

4 Experimental Results

4.1 The Iris Video Dataset

In order to evaluate the performance of the four methods for iris image quality assessment, we selected the Multiple Biometrics Grand Challenge “MBGC.v2” database [19]. It was collected during the spring of 2008 by The Computer Vision Research Lab at the University of Notre Dame and provided 986 near infrared eye videos. All videos were acquired using an LG2200 EOU iris capture system [20] (see figure 8a). The camera uses near-infrared illumination of the eye. The iris video sequences were digitized by a DayStar XLR8 USB video digitizer attached to a Macintosh host system and stored in MPEG-4 format. The size for each frame in the video has 480 rows and 640 columns in 8 bits-grayscale space (intensity values between 0 to 255). The MBGC database presents noise factors, especially those relative to reflections, contrast, luminosity, eyelid and eyelash iris obstruction and focus characteristics. These facts make it the most appropriate for the objectives of our work.

We produced our own database of 4432 iris images from the MBGC iris video database v2. 100 videos were randomly selected from this database. Our database contains 2077 clear iris images (positive samples) and 2355 defocused iris images (negative samples). The all 4432 iris images were manually checked and selected in positive and negative samples by a subjective process (based on human perception of defocus). In iris images where it had uncertainty about whether it was a clear iris image or a defocused one, verification tests were implemented. Thus, the negative samples come from those iris images that cannot be segmented by Libor Masek recognition algorithm [21]. The iris images that did not have good iris segmentation were replaced. For instead, iris images that show blinks and off-angle, see figure 8.

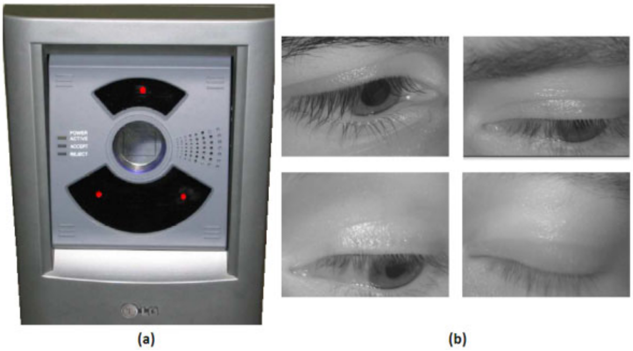


Fig. 8. a) LG EOU2200 system b) The iris images that do not include enough iris information

4.2 Best Iris Image Selection

The table 1, shows the obtained results, the first column identifies the evaluated kernel, the second and third column show the mean and the standard deviation of the total power at high frequency bands processing each kernel with clear iris images database. The next two columns give the mean and the standard deviation processing each kernel with the defocused iris images database.

The first last column indicates the optima's threshold for discrimination of defocus iris images. We use the receiver operation characteristic (ROC) curves to obtain the optimal decision threshold.

Table 1. Results of the evaluated kernel methods

Kernel	Clear iris images		Defocus iris images		Optimal Decision threshold
	Mean	Standard deviation	Mean	Standard deviation	
Daugman	55.063	8.9134	26.838	5.7642	**39.9490**
Wei et al.	19.928	4.5584	12.291	5.3759	**14.1631**
LoG	108.49	10.651	72.88	14.658	**92.0776**
Kang & Park	25.792	6.0658	13.25	3.9636	**15.8247**

The error percentages for every kernel are shown in the table 2. If an accepted/positive iris image (focus image) is a defocus image, it is called a false accept. The percentage of false accepts is called false accept rate (FAR). If a rejected iris image (defocus image) is a focus image, it is called a false reject. The percentage of false reject is called false reject rate (FRR). The best performance scores (minimum FAR and FRR) was generated by the kernel proposed by Kang & Park followed by the Daugman kernel who presented also low error rates. The worse results performance kernels were presented by kernels proposed in [15,16], these kernels had the highest error rates.

Table 2. Error percentages of evaluated kernel methods

Kernel	FAR (%)	FRR (%)
Daugman	**2.8**	**3.6**
Wei et al.	**8.3**	**2.6**
LoG	**3.7**	**5.2**
Kang & Park	**1.6**	**2.3**

To compare the convolution kernels in terms of speed, we compute the total multiplication count (TMC) [14]. For the Daugman's convolution kernel: TMC = (8x8x640/4 x 480/4) = 1228800. For the Laplacian of Gaussian Convolution kernel: TMC = (9x9x640/4 x 480/4) = 1555200. For the convolution kernel of Wei et al and convolution kernel of Kang & Park: TMC = (5x5x640/3 x 480/3) = 853333. With this, it is shown that the last two convolution kernels are 30.56% faster than the Daugman's convolution kernel.

5 Conclusions

In this paper, we present the results of a comparative analysis of four representative convolution kernels for iris image quality assessment to select clear images in the video sequence. The defocus and motion blur assessment allow distinguishing between the images with high quality (clear images) and those with low quality (blurred images). To distinguish between both images classes, an optimal threshold was established. It was experimentally obtained analyzing the database presented in section 4.1. We used the ROC curves to obtain the optimal decision threshold.

From the experimental results, it was concluded that the Kang & Park convolution kernel was superior to the other three kernels in terms of speed and accuracy. The results obtained by this kernel showed the lowest error rates (FAR=1.6% and FRR=2.3%). The results suggest that, to add a quality assessment stage to the video iris recognition system could increase the system performance.

Acknowledgment. This work was supported by SIP2011 grants from IPN.

References

1. Gamassi, M., Lazzaroni, M., Misino, M., Piuri, V.: Quality assessment of biometric systems: a comprehensive perspective based on accuracy and performance measurement. IEEE Transactions on Instrumentation and Measurement 54, 1489–1496 (2005)
2. Chen, Y., Dass, S.C., Jain, A.K.: Localized Iris Image Quality Using 2-D Wavelets. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 373–381. Springer, Heidelberg (2005)
3. Kalka, N.D., Zuo, J., Schmid, N.A., Cukic, B.: Image quality assessment for iris biometric. In: SPIE: Biometric Technology for Human Identification III, vol. 6202, pp. D1–D11 (2006)

4. Cao, Y., Wang, Z., Lv, Y.: Genetic Algorithm Based Parameter Identification of Defocused Image. In: ICCCSIT 2008: International Conference on Computer Science and Information Technology, pp. 439–442 (September 2008)
5. Zuo, J., Schmid, N.A.: Global and local quality measures for NIR iris video. cvprw. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, cvprw, pp. 120–125 (2009)
6. Zhu, X.-D., Liu, Y.-N., Ming, X.: A quality evaluation method of iris images sequence based on wavelet coefficients in region of interest. In: CIT 2004: Proceedings of the The Fourth International Conference on Computer and Information Technology, pp. 24–27. IEEE Computer Society, Washington, DC, USA (2004)
7. Chen, J., Hu, G.S., Xu, J.: Iris: Image Quality Evaluation Method Based on Wavelet Packet Decomposition. Journal oj Tsinghua University (Sci&Tech), 43(3), 377–380 (2003)
8. Ma, L., Tan, T., Wang, Y.: Personal identification based on iris texture analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1519–1533 (2003)
9. Zhang, G., Salganicoff, M.: Method of measuring the focus of close-up image of eyes. Tech. Rep. 5953440, United States Patent (1999)
10. Belcher, C., Du, Y.: Information distance based selective feature clarity measure for iris recognition. In: Proc. SPIE, vol. 8, pp. 64940E1–64940E12 (2007)
11. Tenenbaum, J.M.: Accommodation in computer vision. Ph.D. thesis, Stanford University (1970)
12. Jarvis, R.A.: Focus optimization criteria for computer image processing. Microscope 24(2), 163–180 (1976)
13. Nayar, S.K., Nakagawa, Y.: Shape from focus. IEEE Trans. Pattern Analysis and Machine Intelligence 16, 824–831 (1994)
14. Daugman, J.G.: How Iris Recognition Works. IEEE Trans. on Circuits and Systems for Video Technology 14(1), 21–30 (2004)
15. Wei, Z., Tan, T., Sun, Z., Cui, J.: Robust and Fast Assessment of Iris Image Quality. In: Zhang, D., Jain, A.K. (eds.) ICB 2005. LNCS, vol. 3832, pp. 464–471. Springer, Heidelberg (2005)
16. Jing, W., He, X.F., Pengfei, S.: An Iris Image Quality Assessment Method Based on Laplacian of Gaussian Operation. In: MVA 2007 IAPR Conference on Machine Vision Applications, Tokyo, Japan, May 16-18, pp. 248–251 (2007)
17. Kang, B.J., Park, K.R.: A Study on Iris Image Restoration. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 31–40. Springer, Heidelberg (2005)
18. Burges, J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
19. Multiple Biometric Grand Challenge, `http://face.nist.gov/mbgc/`
20. Bowyer, K.W., Hollingsworth, K., Flynn, P.J.: Image Understanding for Iris Biometrics: a Survey. Computer Vision and Image Understanding 110(2), 281–307 (2008)
21. Masek, L.: Recognition of human iris patterns for biometric identification. Master's thesis, University of Western Australia (2003)

Optimization of Parameterized Compactly Supported Orthogonal Wavelets for Data Compression

Oscar Herrera Alcántara[1] and Miguel González Mendoza[2]

[1] Departamento de Sistemas, Universidad Autónoma Metropolitana Azcapotzalco
Av. San Pablo 180, Col. Reynosa Tamaulipas, 02200, México, D.F.
oha@correo.azc.uam.mx

[2] División de Ingeniería y Arquitectura, ITESM-CEM
Carretera al Lago de Guadalupe, Km 3.5, Atizapán, 52926, México, México
mgonza@itesm.mx

Abstract. In this work we review the parameterization of filter coefficients of compactly supported orthogonal wavelets used to implement the discrete wavelet transform. We also present the design of wavelet based filters as a constrained optimization problem where a genetic algorithm can be used to improve the compression ratio on gray scale images by minimizing their entropy and we develop a quasi-perfect reconstruction scheme for images. Our experimental results report a significant improvement over previous works and they motivate us to explore other kinds of perfect reconstruction filters based on parameterized tight frames.

Keywords: Wavelets, parameterization, optimization, data compression, entropy, genetic algorithm.

1 Introduction

Wavelets are families of functions $\psi_{a,b}(x) = a^{\frac{1}{2}}\psi(ax-b)$ with $a \neq 0$, associated to scaling functions that have the property of being described as linear combinations of translated and dilated versions of themselves. In particular when $a = 2$ it is called dyadic scaling. The wavelet function $\psi(x)$ is orthonormal if the family $\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi(2^j x - k)$ is an orthonormal basis of the Hilbert space $L^2(\mathbb{R})$, and the support of $\psi : \Omega \to \mathbb{R}$ is $supp\ \psi = \{x \in \Omega | \psi(x) \neq 0\}$. Scaling functions $\varphi(x)$ give place to a multi-resolution analysis (MRA) that consists of the decomposition of $L^2(\mathbb{R})$ into the chain of closed subspaces $\cdots V_{j-1} \subset V_j \cdots \subset V_{j+1} \cdots$, where $V_j = Span\{\varphi_{jk}(x),\ k \in \mathbb{Z}\}$ and such that $\cap_j V_j = \{0\}$, $\cup_j V_j = L^2(\mathbb{R})$, so it aims to decompose $L^2(\mathbb{R})$ as $V_{j0} \oplus \sum_{j \geq j_0} W_j$, where W_j is defined as the orthogonal complement of V_j in V_{j+1} that is $V_{j+1} = V_j \oplus W_j$. Compactly supported orthogonal wavelets can be obtained from scaling functions through the dilation equation:

$$\varphi(x) = \sum_{k=0}^{N-1} h_k \varphi(2x-k) \tag{1}$$

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 510–521, 2011.

where $N \in \mathbb{N}$, $h_k \in \mathbb{R}$, and $h_k = 0$ for $k < 0$ and $k \geq N$. A filter F is determined by its impulse response, which is a finite sequence of length N. In this case we say that F has length N. From now on, we will use filter or sequence equivalently. As a simple example, consider the finite sequence of length two $h_k = \{1, 1\}$ for the Haar scaling function and $g_k = \{1, -1\}$ for the Haar wavelet function. The sequences h_k and $g_k = (-1)^k h_{1-k}$, for $k = 0, 1, \ldots, N-1$ represent the impulse response of a filter bank formed by a low pass filter and a high pass filter, that can be used to implement the discrete wavelet transform [1]. Typically, there is no closed analytic form for the scaling and wavelet functions but the cascade algorithm [2] can be used to plot them from h_k and g_k. There are several approaches to construct and to choose wavelets sequences for different applications. In previous works the construction or selection have been based on the user's experience, ease of use, smoothness, or known libraries of wavelets [3]. In particular, in her seminal work, Daubechies [4] constructed orthonormal bases satisfying the constraints for the h_k sequences that imposes the maximization of the number of wavelet vanishing moments. The N vanishing moments are described by:

$$M_p = \int x^p \varphi(x) dx, p = 0, 1, 2, \ldots, N \tag{2}$$

As p grows, the wavelet will be smoother. If the M_p moments are lower than a threshold value, they are considered negligible and the decomposition of a signal concentrates its power in a reduced number of coefficients. This property is desirable from the point of view of compression (our main goal) because that means that we are able to find appropriate wavelet bases for the representation of the analyzed function. In our research we do not use a set of predefined wavelets for data compressing but we aim to construct optimal wavelets for each signal to be processed.

The parameterization of the h_k and g_k sequences [5][6][7][8][9] plays an important role in the design of wavelets once it reduces the search space of solutions, and predetermines the wavelet family characteristics. Several works report that there is no unique parameterization, so when proposing or choosing a parameterization at least three goals should be considered [10]: to cover all possible orthonormal filters, to determine the intervals for the parameters which permit all wavelets to be found uniquely, and to express the parameterization in such a way that the coefficients can be easily generated. In this work we prefer to use the parameterizations of orthonormal wavelet bases described in [11] to optimize parameterized wavelet sequences of length eight and ten in order to achieve maximum compression of gray scale images. Certainly, this is not the first work in proposing the design of personalized wavelets, in [3] for example a feedback-based approach is used on parameterized wavelets to compress images, but that methodology is less efficient than ours since it requires to apply the wavelet transform, the zero-tree and entropic encoding, followed by the corresponding reverse steps, after this the PNSR is maximized using a genetic algorithm. In our case we do not depend of a specific encoding and no reconstruction steps are required, but we optimize the representation of a given signal in the wavelet space (we minimize the entropy of the transformed signal). With this approach we reduce the execution time considerably. Since we conserve nearly all the energy, we almost achieve perfect reconstruction.

This article is organized as follows: In Section 2 we review the parameterization of filters of length four to ten and discuss some properties of them. In Section 3 we comment on the possibility of applying the energy conservation principle to optimize the free parameters that describe the h_k and g_k filters, so we are able to design personalized wavelets for image compression. In Section 4 we present how a genetic algorithm can be used to deal with the constrained optimization problem proposed in the previous section, and in Section 5 we present experimental results. Finally in Section 6 we present conclusions and future work.

2 Parameterization of Filter Coefficients

In this section we review the parameterization of filters of length four to ten. The simplest filter of length two has no parameters and it corresponds to the Haar wavelet. For larger sequences we consider the dilation equation (1) and the conditions (3) and (4) that are necessary for uniqueness of a solution and to satisfy the orthonormality criterion, that is:

$$\sum_{k\in\mathbb{Z}} h_k = 2 \tag{3}$$

and

$$\sum_{k\in\mathbb{Z}} h_k h_{k+2l} = 2\delta_{0l} \quad (l \in \mathbb{Z}) \tag{4}$$

where δ_{0l} is the discrete Kronecker delta function. In the z-domain, if $|z| = 1$, there is equivalence between (4) and (5) so:

$$|H(z)|^2 + |H(-z)|^2 = 1 \tag{5}$$

where $H(z)$ is the so called trigonometric polynomial given by:

$$H(z) = \sum_{k} h_k z^k. \tag{6}$$

The condition

$$H(z = 1) = 1 \tag{7}$$

is equivalent to (3), and the condition (7) together with (5) implies that:

$$H(z = -1) = 0. \tag{8}$$

The previous conditions define the criterion that can be used to parameterize wavelet based filters, as is described in the next section.

2.1 Parameterization of Filters of Length Four

As an example we review the parameterization of filters of length four. Let $H_4(z) = a_0 + b_0 z + a_1 z^2 + b_1 z^3$ be the corresponding trigonometric polynomial. Condition (5) implies that:

$$2(a_0^2 + a_1^2 + b_0^2 + b_1^2) + 2(a_0 a_1 + b_0 b_1)z^{-1} + 2(a_0 a_1 + b_0 b_1)z^2 = 1. \qquad (9)$$

To ensure a perfect reconstruction bank filter, the first term of (9) must be:

$$a_0^2 + a_1^2 + b_0^2 + b_1^2 = \frac{1}{2}. \qquad (10)$$

At the same time the second and third terms of (9) are equal to zero, so:

$$a_0 a_1 + b_0 b_1 = 0. \qquad (11)$$

Multiplying (11) by -2 and adding to (10) we get a sum of perfect squares:

$$(a_0 - a_1)^2 + (b_0 - b_1)^2 = \frac{1}{2}. \qquad (12)$$

Now it is possible to introduce the parameter[1] $\alpha \in [0, 2\pi)$, hence:

$$(a_0 - a_1) = \frac{1}{\sqrt{2}} cos\alpha \qquad (13)$$

and

$$(b_0 - b_1) = \frac{1}{\sqrt{2}} sin\alpha. \qquad (14)$$

Additionally, considering the equation (7) with $H_4(z)$ we obtain:

$$a_0 + b_0 + a_1 + b_1 = 1 \qquad (15)$$

and considering (8) with $H_4(z)$ we get:

$$a_0 - b_0 + a_1 - b_1 = 0. \qquad (16)$$

Adding (15) to (16) we get:

$$a_0 + a_1 = \frac{1}{2}. \qquad (17)$$

Subtracting (16) to (15) we get:

$$b_0 + b_1 = \frac{1}{2}. \qquad (18)$$

Finally, combining (17) and (18) we obtain the next parameterizations for the filters of length four:

$$\begin{array}{ll} a_0 = \frac{1}{4} + \frac{1}{2\sqrt{2}} cos\alpha & b_0 = \frac{1}{4} + \frac{1}{2\sqrt{2}} sin\alpha \\ a_1 = \frac{1}{4} - \frac{1}{2\sqrt{2}} cos\alpha & b_1 = \frac{1}{4} - \frac{1}{2\sqrt{2}} sin\alpha. \end{array} \qquad (19)$$

Following the same procedure is possible to describe parameterizations for larger filters, and to deduce that all orthogonal filters have even length.

[1] It has been reported that all parameterizations of the filter coefficients expressed in terms of trigonometric functions have no natural interpretation of the angular parameters for the resulting scaling function.

Special Cases for Filters of Length Four

Case of $\alpha = \frac{\pi}{4}$. In this case the trigonometric polynomial $H_4(z)$ is associated with the Haar wavelet: $a_0 = \frac{1}{2}; b_0 = \frac{1}{2}; a_1 = 0$; $b_1 = 0$, the low pass filter is $h_k = \{\frac{1}{2}, \frac{1}{2}, 0, 0\}$ and the high pass filter is $g_k = \{\frac{1}{2}, -\frac{1}{2}, 0, 0\}$.
Case of $\alpha = \frac{5\pi}{12}$. The vanishing moment condition means:

$$\frac{d}{dz}H(z)|_{z=-1} = b_0 - 2a_1 + 3b_1 = 0 \tag{20}$$

which is only satisfied by $\alpha = \frac{5\pi}{12}$ or $\frac{13\pi}{12}$. For the first case $a_0 = \frac{1+\sqrt{3}}{8}$, $a_1 = \frac{3+\sqrt{3}}{8}$, $b_0 = \frac{3+\sqrt{3}}{8}$, $a_1 = \frac{1-\sqrt{3}}{8}$ that are associated with the Daubechies 4 (Dau4) wavelet.

Parameterization of Filters of Length Sixth

Following a similar approach to the filters of length four the next parameterizations for the filters of length sixth are obtained:

$$\begin{array}{ll} a_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}}cos\alpha + \frac{p}{2}cos\beta\ , & b_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}}sin\alpha + \frac{p}{2}sin\beta \\ a_1 = \frac{1}{4} - \frac{1}{2\sqrt{2}}cos\alpha \qquad\qquad , & b_1 = \frac{1}{4} - \frac{1}{2\sqrt{2}}sin\alpha \\ a_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}}cos\alpha - \frac{p}{2}cos\beta\ , & b_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}}sin\alpha - \frac{p}{2}sin\beta \end{array} \tag{21}$$

where $p = \frac{1}{2}\sqrt{1 + sin(\alpha + \frac{\pi}{4})}$, $\alpha, \beta \in \mathbb{R}$. A set of labelled special cases are shown in Table 1. Note that eight digits are reported for each parameter according to the precision of the encoding used in the experiments (see Section 5).

Table 1. Special cases of parametric filters of length sixth

α	β	p	Name
0.78539816	0.78539816	0.70710678	Haar
1.30899694	1.04719755	0.68301270	Dau4
1.78508070	1.07424683	0.62059838	Dau6

Parameterization of Filters of Length Eight

For this case $H_8(z) = a_0 + b_0 z + a_1 z^2 + b_1 z^3 + a_2 z^4 + b_2 z^5 + a_3 z^6 + b_3 z^7$ and the parameterizations of the filters of length eight are:

$$\begin{array}{ll} a_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}}cos\alpha + \frac{1}{2\sqrt{2}}cos\beta cos\gamma & b_0 = \frac{1}{8} + \frac{1}{4\sqrt{2}}sin\alpha + \frac{1}{2\sqrt{2}}sin\beta cos\theta \\ a_1 = \frac{1}{8} - \frac{1}{4\sqrt{2}}cos\alpha + \frac{1}{2\sqrt{2}}cos\beta sin\gamma & b_1 = \frac{1}{8} - \frac{1}{4\sqrt{2}}sin\alpha + \frac{1}{2\sqrt{2}}sin\beta sin\theta \\ a_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}}cos\alpha - \frac{1}{2\sqrt{2}}cos\beta cos\gamma & b_2 = \frac{1}{8} + \frac{1}{4\sqrt{2}}sin\alpha - \frac{1}{2\sqrt{2}}sin\beta cos\theta \\ a_3 = \frac{1}{8} - \frac{1}{4\sqrt{2}}cos\alpha - \frac{1}{2\sqrt{2}}cos\beta sin\gamma & b_3 = \frac{1}{8} - \frac{1}{4\sqrt{2}}sin\alpha - \frac{1}{2\sqrt{2}}sin\beta sin\theta \end{array} \tag{22}$$

Subject to:

$$\begin{array}{l} \sqrt{2}cos\theta sin\beta - 2cos\theta sin\alpha sin\beta + \sqrt{2}cos\beta(cos\gamma - sin\gamma) \\ -4cos^2\beta cos\gamma sin\gamma - 2cos\alpha cos\beta(cos\gamma + sin\gamma) \\ -\sqrt{2}sin\beta sin\theta - 2sin\alpha sin\beta sin\theta - 4cos\theta sin^2\beta sin\theta = 0 \end{array} \tag{23}$$

As the case of filters of length four and sixth, a set of labelled special cases are shown in Table 2, now for the filters of length eight.

Table 2. Special cases of parametric filters of length eight

α	β	γ	θ	Name
0.78539816	0.78539816	0	0	Haar
1.30899694	1.00931436	0.43451875	6.12970142	Dau4
1.78508069	0.90000966	0.83293201	6.109977902	Dau6
2.24007543	0.75354199	0.96140244	6.25777293	Dau8

Parameterization of Filters of Length Ten

The parameterizations for the filters of length ten are:

$$\begin{aligned}
a_0 &= \tfrac{1}{16} + \tfrac{1}{8\sqrt{2}} cos\alpha + \tfrac{1}{4\sqrt{2}} cos\beta cos\gamma + \tfrac{r}{2} cos\delta \\
b_0 &= \tfrac{1}{16} + \tfrac{1}{8\sqrt{2}} sin\alpha + \tfrac{1}{4\sqrt{2}} sin\beta cos\theta + \tfrac{r}{2} sin\delta \\
a_1 &= \tfrac{1}{8} - \tfrac{1}{4\sqrt{2}} cos\alpha + \tfrac{1}{2\sqrt{2}} cos\beta sin\gamma \\
b_1 &= \tfrac{1}{8} - \tfrac{1}{4\sqrt{2}} sin\alpha + \tfrac{1}{2\sqrt{2}} sin\beta sin\theta \\
a_2 &= \tfrac{1}{8} + \tfrac{1}{4\sqrt{2}} cos\alpha - \tfrac{1}{2\sqrt{2}} cos\beta cos\gamma \\
b_2 &= \tfrac{1}{8} + \tfrac{1}{4\sqrt{2}} sin\alpha - \tfrac{1}{2\sqrt{2}} sin\beta cos\theta \\
a_3 &= \tfrac{1}{8} - \tfrac{1}{4\sqrt{2}} cos\alpha - \tfrac{1}{2\sqrt{2}} cos\beta sin\gamma \\
b_3 &= \tfrac{1}{8} - \tfrac{1}{4\sqrt{2}} sin\alpha - \tfrac{1}{2\sqrt{2}} sin\beta sin\theta \\
a_4 &= \tfrac{1}{16} + \tfrac{1}{8\sqrt{2}} cos\alpha + \tfrac{1}{4\sqrt{2}} cos\beta cos\gamma - \tfrac{r}{2} cos\delta \\
b_4 &= \tfrac{1}{16} + \tfrac{1}{8\sqrt{2}} sin\alpha + \tfrac{1}{4\sqrt{2}} sin\beta cos\theta - \tfrac{r}{2} sin\delta
\end{aligned} \tag{24}$$

where: $r = \sqrt{\frac{1}{2} - a_1^2 - a_2^2 - a_3^2 - b_1^2 - b_2^2 - b_3^2}$, and subject to:

$$\begin{aligned}
&cos\beta(cos\gamma(\sqrt{2} - 2cos\alpha) - 8\sqrt{2} r cos\delta sin\gamma) + \\
&sin\beta(cos\theta(\sqrt{2} - 2sin\alpha) - 8\sqrt{2} r sin\delta sin\theta) = 0
\end{aligned} \tag{25}$$

Special cases are shown in Table 3. An important property of parameterizations described in (19), (21), (22) and (24) is that the parameterization of shorter sequences is included in the parameterization of larger sequences. In the next

Table 3. Special cases of parametric filters of length ten

α	β	γ	θ	δ	Name
0.78539816	0.78539816	0	0	785398163	Haar
1.30899694	1.00931436	0.43451875	6.12970145	1.04719755	Dau4
1.78508069	0.90009666	0.83293201	6.10997785	1.17974313	Dau6
2.24007771	0.75354092	0.96140314	6.25777331	1.25902993	Dau8
2.68294774	0.71493948	1.02836200	0.22250858	1.29078304	Dau10

sections, we discuss how to establish the design of wavelets for data compression as an optimization of the constrained parameterized sequences, and how to define the objective function of an evolutionary algorithm.

3 Wavelet Image Processing and Energy Conservation

An image is a two-dimensional function $f(x, y)$ with discrete values $x \in [0, w]$ and $y \in [0, h]$, where w is the horizontal dimension or width, and h is the vertical dimension or height. Each element at position (x, y) is called a pixel. In case of grayscale images $f(x, y)$ takes values in $[0, 255]$, where a zero value corresponds to a black pixel, whereas the maximum value 255 is a white pixel. The two-dimensional discrete wavelet transform is calculated applying a horizontal transformation step followed from a vertical transformation step. This gives us a pyramidal effect that gives places to four subbands HH, HL, LH and LL, with $\frac{w}{2}$ x $\frac{h}{2}$ wavelet coefficients w_i [12]. The HH subband stores the most detailed information of the image, and the LL subband stores the coarser information. A second transformation level is applied to the LL subband, that generates other four subbands, and the process continues while the width and height of the new subbands are larger than the filter length. The two-dimensional inverse wavelet transform is calculated in reverse order, starting at the top level of the pyramidal subbands. The reconstruction introduces floating point operations, precision errors and filtering errors that can be measured with the RMS error (or the PNSR) comparing the original with the reconstructed image. The energy of a discrete signal with N samples is given by:

$$E = \sum_{i=0}^{N-1} f_i^2. \tag{26}$$

The cumulative energy of the signal is given by:

$$E_c = \sum_{i=0}^{r} f_i^2, \quad 0 \leq r \leq N. \tag{27}$$

After applying the filter step in the discrete wavelet transform the wavelet coefficients w_i must conserve the energy, so:

$$E = \sum_{i=0}^{N-1} w_i^2 \tag{28}$$

Note that, as a simple example, the Haar sequences $h_k = \{1, 1\}$ and $g_k = \{1, -1\}$ do not conserve the energy but choosing an appropriated scaling factor we obtain the sequences $h_k = \{2^{-\frac{1}{2}}, 2^{-\frac{1}{2}}\}$ and $g_k = \{2^{-\frac{1}{2}}, -2^{-\frac{1}{2}}\}$ which indeed conserve the energy. The same occurs for the rest of the parameterized sequences multiplied by the scaling factor $2^{-\frac{1}{2}}$. The higher the concentration of energy in the lowest number of coefficients, the higher the quality of the filter and the number of negligible coefficients, and consequently it is more useful to compress a given signal [13]. We can sort the transformation coefficients from largest to smallest and measure the cumulative energy. When it reaches an acceptable value of the total, the remaining coefficients can be ignored, and this determines an automatic threshold (T_h) for the coefficients that will be forced to be zero.

4 Genetic Algorithm and the Constrained Optimization Problem

As we point out, a peculiarity of the transformation coefficients when a good filter is used is that a small number of them concentrates the most of the energy, then a way to measure the efficiency of a filter is counting the number of coefficients which lie below the T_h value (negligible coefficients) and that do not change substantially the cumulative energy. Considering this, we propose to use a genetic algorithm to maximize the fitness function:

$$fitness = NegligibleCoefficients \tag{29}$$

where as already mentioned, NegligibleCoefficients (NC) is the number of (near) zero wavelet coefficients using the corresponding parameterized filter. It has been shown that for filters of length eight or larger there are non-linear constraints that must be satisfied, and to deal with them we use the penalty criterion [14] where given C constraints, a large number K is chosen and for each non-satisfied restriction an individual of the genetic algorithm receive a penalty $\frac{r}{C}$, $0 \leq r \leq C$. The modified fitness function that includes the penalty criterion is:

$$fitness = NegligibleCoefficients - K(1 - \frac{r}{C}) \tag{30}$$

where r is the number of fulfilled constraints. When the individuals satisfy all the restrictions ($r = C$) the fitness function of the genetic algorithm is reduced to the non-constrained case.

In this work we process grayscale images which are typical in image processing. The image dimensions are 64×64 pixels that give us a total of 4096 wavelet transform coefficients. Our goal is to discover the best h_k sequences by optimizing the parameters of the equations described in section 2 for the filters of length eight and ten. For this purpose we appeal to a genetic algorithm [15] characterized by:

- A population of even size P, with a temporary population of size $2P$.
- Pair wise for crossover between the i-th and the $(P - 1 - i)$-th sorted individuals, $i \in [0, P1]$.
- Deterministic selection, the best P individuals from the temporal population.
- Annular crossover.

Each individual models the values of the free parameters in the interval $[0, 2\pi)$, and the fitness function is described in (30). The threshold value Th is automatically calculated after the transformed coefficients have been sorted in descending order of magnitude, and the cumulative energy divided by the total energy is higher than 0.999999.

5 Experimental Results

We developed three experiments:

Experiment 1. We calculate the correlation between the entropy of the wavelet coefficients and the number of non-negligible coefficients. The samples were calculated with a filter of length four and testing with $\alpha \in [0, 2\pi)$ with increment of 0.01 for the lena image. The correlation value was 0.93 (close to 1.0) and supports the idea that if we increase the number of negligible coefficients, the entropy decreases and the compression ratio will increase by applying an entropic coder as states the Shannon's source coding Theorem [16].

Experiment 2. We maximize the number of negligible coefficients (NC) by optimizing the free parameters of filters of length eight and using a genetic algorithm under the next conditions: Number of generations $G = 60$, Probability of mutation $Pm = 0.03$, Probability of crossover $Pc = 0.97$, Population size $P = 16$ individuals and $K = 1000000$. Each parameter was encoded with 29 bits (weighted binary) after an experimental analysis about the required precision of the filter coefficients. Consider that the individuals of the genetic algorithm initialized randomly could not to satisfy the constraint (23), and to prevent a wrong solution, we initialize four individuals with the corresponding parameters of Haar, Dau4, Dau6, and Dau8 filters (see Table 2) which of course satisfy all the restrictions, so in the worst case one of them will be the solution. The evolutionary optimization process to compress an image require $2PG = 1920$ two-dimensional wavelet transformations. Consider that the wavelet transform for a vector with n components has $O(n)$ complexity. An advantage of our methodology is that we do not require applying the inverse wavelet transform because the filters provide perfect reconstruction[2].

In the experiments, the average of the execution time required for the VGA to optimize the parameters was 165.3 seconds, running on a Debian/Linux PC, with 2 GB RAM and Intel Celeron 2.2 GHz processor. The reconstruction RMS error for the images Baboon, Barbara, Clegg, Frymire, Lena, Monarch, Peppers and Sail has an average value of 0.014 with standard deviation of 0.003, so we achieve a quasi-perfect reconstruction in spite of the large number of negligible coefficients.

In Table 4 we compare the number of negligible coefficients (higher is better) between our evolutionary filter of length four (Param8) and Haar (1st row), Dau4 (2nd row), Dau6 (3rd row), Dau8 (4th row), and the so called Evolets (5th row) reported in [17] that uses a single parameter. As we can appreciate, in all cases the optimized filters provides an evident better performance than the already published. In fact, our optimized sequences reach 2.36 -average- times the number of maximum negligible coefficients of the other filters on the same processed images. In all cases the constraint described in (23) was satisfied with a precision less than 10^{-4} and the average of this was 0.00099 with a standard deviation of 0.000004.

[2] In the experiments the inverse transform was calculated to report the RMS error, but this step is not necessary.

Experiment 3. We optimize the parameters of filters of length ten with $Pc = 0.97$, $P = 40$ individuals, $K = 1000000$, 30 bits of precision, and Pm=0.01 for the barbara and sail images and $Pm = 0.02$ for the rest of the images. Five individuals were initialized with the corresponding parameters of Haar, Dau4, Dau6, Dau8 and Dau10 of Table 3. First, we test with a maximum of $G = 90$ generations, that required about 634 seconds, and we note that each image has a different convergence. Then we observe that after an average of 66 generations there is no improvement and in this case the average execution time was 448 seconds. The results of the maximum reached negligible coefficients are reported at the last row (Param10) of Table 4. See, for example, how in the case of the images lena, monarch and peppers the number of negligible coefficients is taken over 30%.

Table 4. Experimental results: optimization of filters for images 1. Baboon, 2. Barbara, 3. Clegg, 4. Frymire, 5. Lena, 6. Monarch, 7. Peppers and 8. Sail.

Image	Baboon	Barbara	Clegg	Frymire	Lena	Monarch	Peppers	Sail
Filter	NC	NC	NC	NC	NC	NC	NC	NC
Haar	214	327	272	373	419	413	371	228
Dau4	220	348	251	279	464	567	392	240
Dau6	261	359	251	231	443	595	385	231
Dau8	215	320	263	227	433	594	391	216
Evolets4	244	361	289	365	469	580	404	256
Param8	661	1052	710	706	1168	1053	1215	575
Param10	717	1052	711	793	1232	1297	1239	575

In Table 5 and 6 we report the best evolutionary filter for the processed images that provide the results shown in Table 4. With these values it is possible to plot the corresponding scaling and wavelet functions using the cascade algorithm [2] but they are not reported here in order to save space. For the same reason, the precision of the filter coefficients were truncated but they are available on request.

6 Conclusions and Future Work

The parameterization of filters provides a way to design families of personalized wavelet functions for specific applications and signal processing. The problem of choosing a good wavelet for image compression has been stated as an optimization problem where the energy conservation principle plays an important role. The parameters can be optimized with an evolutionary algorithm to concentrate the signal energy in the lowest number of relevant coefficients and to minimize the entropy of the transformed image. As the number of free parameters increases with the length of the filter, the execution time also does, however it provides a greater freedom degree in the design of wavelets and the number of

Table 5. Filters of length eight optimized with the GA for babbon (1st row), barbara (2nd row), clegg (3rd row), frymire (4th row), lena (5th row), monarch (6th row), peppers (7th row) and sail (8th row)

Baboon	Barbara	Clegg	Frymire	Lena	Monarch	Peppers	Sail
0.482978	0.836497	0.224206	-0.129374	-0.000003	0.000017	-0.000073	-0.000032
0.332584	0.806886	0.459979	-0.134857	-0.085542	0.035191	0.000086	-0.000113
0.71267	0.70148	-0.001174	0.00126	-0.003194	0.003241	-0.001196	0.001126
-0.000228	0.000082	0.000233	0.000079	0.482728	0.836559	0.224374	-0.129612
0.709494	0.704705	-0.000121	0.000208	-0.002116	0.00213	-0.000151	0.000063
0.707435	0.706778	-0.000253	0.000342	0.000251	-0.000252	-0.000328	0.000239
0.23031	0.714754	0.631035	-0.027949	-0.186967	0.030935	0.032728	-0.010633
-0.000166	0.000191	0.000044	0.000689	0.48303	0.836352	0.224198	-0.130125

Table 6. Filters of length ten optimized with the GA for babbon (1st row), barbara (2nd row), clegg (3rd row), frymire (4th row), lena (5th row), monarch (6th row), peppers (7th row) and sail (8th row)

Baboon	Barbara	Clegg	Frymire	Lena	Monarch	Peppers	Sail
0.332788	0.332791	0.707465	0.708838	0.332789	0.160108	0.332789	0.730021
0.806997	0.806959	0.706748	0.705359	0.806954	0.603955	0.806973	0.683033
0.459624	0.459692	-0.000066	-0.001813	0.459701	0.724205	0.459671	-0.000183
-0.135071	-0.135049	0.000154	0.001926	-0.135048	0.13846	-0.135055	0.00027
-0.08517	-0.085255	-0.000491	0.001075	-0.085261	-0.242211	-0.08523	-0.011043
0.035432	0.035268	0.000492	-0.001092	0.035276	-0.032393	0.035274	0.011412
-0.000496	-0.000189	-0.00007	-0.001826	-0.000198	0.077675	-0.000215	-0.000192
-0.000103	-0.000043	-0.000018	0.001751	-0.000045	-0.006273	-0.000047	0.000106
0.000361	0.000069	0.000269	0.000833	0.000077	-0.012669	0.000091	-0.011496
-0.000149	-0.000028	-0.000269	-0.000837	-0.000032	0.003359	-0.000038	0.012287

negligible coefficients is noticeably enhanced. The execution time of our method is still too large to be used in a real-time encoding scheme and it represents a challenge in our research. To improve on this drawback we pretend to explore other evolutionary algorithms and the use of FPGA technologies. The current results motivate us to work with the parameterization of wavelet tight frames (that encloses the orthogonal filters as a particular case, see [18]) so we get larger freedom degrees. Additionally, we pretend to optimize the parameters for each transformation level, once it has been reported that it is possible, and we will work on this item. At the moment, we present an offline encoding scheme with quasi-perfect reconstruction that can be used in digital encyclopedias, CD/DVD storage, and other non-symmetrical systems where the encoding process is performed once but the decoding is performed by a large number of users. The encoding scheme offers an alternative to lossless compression methods where the compression rates are limited by the high entropy of the images. Furthermore, as our method admits the setting of the preserved energy (in the experiments $\frac{Ec}{E} = 0.999999$), it can be used as a quality factor for the reconstructed image, in a similar way to JPEG with the Q-factor[19]. The applications of parametric filters are not restricted to image compression, so we propose to design custom filters for other image processing tasks such as edge detection, segmentation, and enhancement, among others.

References

1. Mallat, S.G.: Multiresolution approximations and wavelet orthonormal bases of $\mathbb{L}^2(\mathbb{R})$. Trans. Amer. Math. Soc. 315(1), 69–87 (1989)
2. Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. on Patt. Anal. Mach. Intell. 11(7), 674–693 (1989)
3. Hereford, J.M., Roach, D.D.W., Pigford, R.: Image compression using parameterized wavelets with feedback. In: Proc. of SPIE, vol. 5102, pp. 267–277 (April 2003)
4. Daubechies, I.: Orthonormal bases of compactly supported wavelets. Commun. on Pure and Applied Math. 41(1), 909–996 (1988)
5. Hehong, Z., Tewfik, A.: Parametrization of compactly supported orthonormal wavelets. IEEE Transactions on signal processing 41(3), 1428–1431 (1993)
6. Pollen, D.: Su1(2f(z,1/z) for f a subfield of c. J. Amer. Math. Soc. 3, 611–624 (1990)
7. Wells, R.O.: Parametrizing smooth compactly supported wavelets. Trans. Amer. Math. Soc. 338(2) (August 1993)
8. Schneid, J., Pittner, S.: On the parametrization of the coefficients of dilation equations for compactly supported wavelets. Computing 51, 165–173 (1993)
9. Lina, J.M., Mayrand, M.: Parametrizations for daubechies wavelets. Ohys. Rev. E 48(6), R4160–R4163 (1993)
10. Sherlock, B.G., Monro, D.M.: On the space of orthonormal wavelets. IEEE Transactions on Signal Processing 46(6), 1716–1720 (1998)
11. Ming-Jun, L., Roach, W.: Parameterization of univariate orthogonal wavelets with short support. Aproximation theory X, Vanderbilt Univ. Press (2002)
12. Mallat, S.: A wavelet tour of signal processing. Academic Press Inc. (1998)
13. Walker, J.: A primer on wavelets and their scientific applications, 2nd edn. Chapman Hall/CRC, Taylor and Francis Group (2008)
14. Coello, C., Lamont, G., Van Velduizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, Heidelberg (2007)
15. Kuri, A.: A Comprehensive Approcah to Genetic Algorithms in Optimization and Learning. National Polytechnic Institute, Mexico (1999)
16. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423 (1948)
17. Herrera, O.: On the best evolutionary wavelet based filter to compress a specific signal. In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part II. LNCS (LNAI), vol. 6438, pp. 394–405. Springer, Heidelberg (2010)
18. Wang, H., Peng, L.: Parameterizations of univariate wavelet tight frames with short support. Communications in Nonlinear Science and Numerical Simulation 11, 663–677 (2006)
19. Pennebaker, W., Mitchell, J.: JPEG: Still Image Data Compression Standard. Springer, Heidelberg (1992)

Efficient Pattern Recalling Using a Non Iterative Hopfield Associative Memory

José Juan Carbajal Hernández[1,2] and Luis Pastor Sánchez Fernández[1]

[1] Center of Computer Research – National Polytechnic Institute, Av. Juan de Dios Bátiz S/N,
Col. Nva. Industrial Vallejo, 07738, México D.F., México. 57296000
juancarvajal@sagitario.cic.ipn.mx, lsanchez@cic.ipn.mx
[2] Computer Science Department, National Institute for Astrophysics,Optics and Electronics,
Sta. Ma. Tonanzintla, Puebla 72840, Mexico

Abstract. Actually associative memories have demonstrated to be useful in pattern processing field. Hopfield model is an autoassociative memory that has problems in the recalling phase; one of them is the time of convergence or non convergence in certain cases with patterns bad recovered. In this paper, a new algorithm for the Hopfield associative memory eliminates iteration processes reducing time computing and uncertainty on pattern recalling. This algorithm is implemented using a corrective vector which is computed using the Hopfield memory. The corrective vector adjusts misclassifications in output recalled patterns. Results show a good performance of the proposed algorithm, providing an alternative tool for the pattern recognition field.

Keywords: Associative memory, Hopfield, non iterative, neural networks.

1 Introduction

Associative memories have considerable importance in the developed activities of numerous researchers, mainly in theory, recognition and pattern classification applications [1]. The John Hopfield work is a very important contribution of research in this field [2], due to the Hopfield model works as an associative memory and as a neural network unifying both research fields [3]. Although the Hopfield model is a good autoassociative memory, it has some limitations; while the number of learned patterns and their dimension increases, the time convergences increases too, even there exist certain cases for those patterns where never converge. In order to resolve this gap, some works have been developed in order to eliminate the non convergence and to increase the recalling patterns as [4].

2 Associative Memories

An associative memory (M) is a system of inputs and outputs that relates as follows: $x \rightarrow [M] \rightarrow y$. The input pattern is represented by a column vector denoted as *x* and the output pattern by a column vector *y*. The finality of an associative memory is to

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 522–529, 2011.

restore full patterns from input patterns that can be altered [4], [5], [6].
Each input pattern forms an association with the corresponding output pattern as (x, y). For a positive integer k, the corresponding association will be denoted as (x^k, y^k).

An associative memory **M** is represented by a matrix whose ij^{th} component is m_{ij} (Palm et al., 1997). The **M** matrix is generated by a finite set of associations known as a fundamental set. The set cardinality is denoted as p.

If μ is an index, the fundamental set can be expressed as follows:

$$\{(x^\mu, y^\mu) | \mu = 1,2, \dots, p\} \tag{1}$$

The patterns that built the fundamental set associations are called fundamental patterns. If it holds that $x^\mu = y^\mu \; \forall \; \mu = 1,2, \dots, p$ the memory is autoassociative, otherwise it is heteroassociative. A distorted version of a pattern x^k to be recalled is denoted as $x^{\sim k}$. If a distorted pattern $x^{\sim\omega}$ is computed as an input by the memory **M** ($\omega \in \{1,2,\dots p\}$), and **M** restores the corresponding fundamental output pattern y^ω, then the recovery is perfect. A perfect memory is one that makes perfect recoveries for all fundamental patterns.

Each column vector that represents an input pattern has n components that fall within the set A, where A=\{0,1\}, and each column vector that represents an output pattern has m components that fall within the set A as follows:

$$x^\mu \in A^n \; and \; y^\mu \in A^m, \forall \; \mu = 1,2, \dots, p \tag{2}$$

An associative memory works in two clearly established phases:

1. Learning phase (creation of the associative memory **M**).
2. Recalling phase (operation of the associative memory **M).**

The column vectors that represent the fundamental input/output patterns can be expressed as follows:

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m \quad \forall \; \mu = 1,2, \dots, p \tag{3}$$

3 The Hopfield Associative Memory

The Hopfield model is an autoassociative memory which is symmetric with zeros in the main diagonal matrix [2]. It is a system of input-output that accepts as input patterns $x^\mu \in A^n, A = \{-1, 1\}$ and produces a fundamental set of $\{(x^\mu, x^\mu) \mid \mu=1, 2, \dots, p\}$.

Learning phase
Following rules outline the learning phase for the Hopfield model, incorporating the pattern associations when the learning phase is finished $\{(x^1, x^1), \dots, (x^p, x^p)\}$.

$$m_{ij} = \begin{cases} \sum_{\mu=1}^{p} x_i^{\mu} x_j^{\mu} & if\ i \neq j \\ 0 \quad if & i = j \end{cases} \tag{4}$$

This expression can be interpreted as follows:

$$\boldsymbol{M} = \sum_{\mu=1}^{p} \left[x^{\mu} \left(x^{\mu} \right)^{t} - I \right] = \left[m_{ij} \right]_{n \times n} \tag{5}$$

where I is the identity matrix of $n \times n$ dimension.

Recalling phase

In the recalling, finding the class means to obtain the same input $x^{\omega} \in A^{m}$ in the output of the associative memory. Presenting an input pattern x^{ω} to the Hopfield memory, the current value will change in future times, adjusting the x_i elements according with the comparison result of the following vector [7], [8]:

$$s_i = \sum_{j=1}^{n} m_{ij} x_j \tag{6}$$

with a threshold value usually zero.

Representing the status of the Hopfield memory in the current time t by $x(t)$; therefore the $x_i(t)$ represents the i^{th} value of the x input in the t time and $x_i(t + 1)$ the i^{th} value of the x input in the next time status $(t + 1)$.

Therefore the recalling phase can be determined in three steps as follows:

1) for $t = 0$, $x(t) = x^{\omega}$; it means $x_i(0) = x_i \ \forall \ i \ \in \{1, 2, 3, \dots, n\}$;
2) $\forall i \ \in \{1, 2, 3, \dots, n\}$ the $x_i(t + 1)$ is calculated according with the following expression:

$$x_i(t+1) = \begin{cases} 1 \ \ if \ \sum_{j=1}^{n} m_{ij} x_j(t) > 0 \\ x_i(t) \ if \ \sum_{j=1}^{n} m_{ij} x_j(t) = 0 \\ -1 \ \ if \ \sum_{j=1}^{n} m_{ij} x_j(t) < 0 \end{cases} \tag{7}$$

3) if the $x_i(t + 1) = x_i(t) \ \forall i \ \in \{1, 2, 3, \dots, n\}$, then the process ends, otherwise the process continue as follows: 2 and 3 steps must be executed as many times as the condition $t = \tau$ is computed, which $x_i(\tau + 1) = x_i(\tau) \ \forall i \ \in \{1, 2, 3, \dots, n\}$: the process ends and the recovered pattern is $x_i(\tau)$.

The convergence process described in step 3, indicates that the system arrives to a locally stable point limit in the time *t*. However, there are some patterns combinations that do not converge in the recalling phase, which means that there are patterns were the iterative process never ends [2], [9].

4 Our Proposed Modification in the Hopfield Model

In this section a new algorithm for the Hopfield memory is introduced in order to eliminate the iteration process in the original model. Our purpose is to introduce an algorithm in the recalling phase that corrects the convergence time in the core of the Hopfield model increasing the recalling performance and reducing the burden computing. Our hypothesis is that an additional corrective vector can increase the capacity of good recalling in the Hopfield associative memory. The following concepts define the steps to follows in order to recover the non-iterative patterns; they are well documented and represent a novelty for the actual associative memory performance.

Definition 1. *Let* $A=\{-1,1\}$ *and* $x^{\omega} \in A^n$ *be a pattern, so the* ij^{th} *component of an associative memory M is denoted by* m_{ij}*, then a thresholded vector* u_i *is determined as follows:*

$$u_i = \begin{cases} \sum_{j=1}^{n} \{m_{ij} | m_{ij} < 0\} & if\ x_i(t+1) > 0 \\ 0 & otherwise \end{cases} \tag{8}$$

Definition 2. *Let* u_i *be the thresholded vector, then the corrective vector* z_i *is calculated according the following rule:*

$$z_i = \begin{cases} -1 & if\ \ u_i < 0 \\ 1 & otherwise \end{cases} \tag{9}$$

Definition 3. *Let* $x_i(t+1)$ *be the output pattern of the memory M and* z_i *the corrective vector, then the adjusted output vector* y_i *can be calculated as follows:*

$$y_i = \begin{cases} x_i(t+1) & if\quad x_i(t+1) = x_i(t) \\ z_i & otherwise \end{cases} \tag{10}$$

5 Numerical Results

For a better understanding of the proposed method, this section exemplifies the operation of the Hopfield model using the iterative and non iterative recalling phases.

5.1 Example

Let the fundamental be a set of p = 3 and n = 5. Given the fundamental patterns $\{(x^{\mu}, x^{\mu}) \mid \mu=1, 2, \ldots, p\}$, determinate the Hopfield memory **M** and recover the output patterns using the iterative and non iterative methods. The fundamental associations are expressed as ordered pairs: $\{(x^{1}, x^{1}), (x^{2}, x^{2}), (x^{3}, x^{3})\}$.

$$x^1 = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \qquad x^2 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \qquad x^3 = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Learning phase

For creating the matrix **M** using the p associations of the fundamental set, it is necessary to follow the equation 4 or 5 of section 2:

First association

$$x^1 \cdot (x^1)^t - I = \begin{pmatrix} 0 & -1 & -1 & 1 & -1 \\ -1 & 0 & 1 & -1 & 1 \\ -1 & 1 & 0 & -1 & 1 \\ 1 & -1 & -1 & 0 & -1 \\ -1 & 1 & 1 & -1 & 0 \end{pmatrix}$$

Second Association

$$x^2 \cdot (x^2)^t - I = \begin{pmatrix} 0 & 1 & -1 & 1 & -1 \\ 1 & 0 & -1 & 1 & -1 \\ -1 & -1 & 0 & -1 & 1 \\ 1 & 1 & -1 & 0 & -1 \\ -1 & -1 & 1 & -1 & 0 \end{pmatrix}$$

Third association

$$x^3 \cdot (x^3)^t - I = \begin{pmatrix} 0 & 1 & -1 & -1 & -1 \\ 1 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 & 0 \end{pmatrix}$$

Hopfield memory

$$M = \sum_{\mu=1}^{p} [x^{\mu} \cdot (x^{\mu})^t - I]$$

$$= \begin{pmatrix} 0 & 1 & -3 & 1 & -3 \\ 1 & 0 & -1 & -1 & -1 \\ -3 & -1 & 0 & -1 & 3 \\ 1 & -1 & -1 & 0 & -1 \\ -3 & -1 & 3 & -1 & 0 \end{pmatrix}$$

Once the learning phase is computed and the associative memory **M** is determined, the next step is to compute the recalling phase.

Recalling phase

In this section, examples between iterative and non iterative associative memories are computed in order to describe the differences between models.

Recalling phase for the Hopfield memory consists in computing with the matrix **M** one fundamental input pattern $x^{\omega} \in A^{n}$, and process the recalling steps (1, 2 and 3) in section 2. As output we expect to obtain the class $x^{\omega} \in A^{n}$. Due to paper space limitations, only the first association (x^{1}, x^{1}) recalling results are shown. The reader can easily verify that the whole fundamental set of patterns is completely recalled. Using the previous definitions it is possible to enunciate The Non Iterative Hopfield Memory Recalling Phase.

Step 1. Obtain *x*(*t*) for *t=0*:

$$x(t) = x(0) = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

Step 2. Obtain *x*(*t+1*) using eq. (7):

$$M \cdot x^1(0) = \begin{pmatrix} 0 & 1 & -3 & 1 & -3 \\ 1 & 0 & -1 & -1 & -1 \\ -3 & -1 & 0 & -1 & 3 \\ 1 & -1 & -1 & 0 & -1 \\ -3 & -1 & 3 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ -6 \\ 4 \\ -6 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

Step 3. A comparison between $x_i(t+1)$ and $x_i(t) \; \forall \; i \in \{1,2,3,\ldots n\}$ must be computed. It is clearly seen that the output pattern does not belong to the input pattern; in this case we have to iterate the steps 2 and 3 again, in order to recover the correct output class as follows:

$$M \cdot x^1(1) = \begin{pmatrix} 0 & 1 & -3 & 1 & -3 \\ 1 & 0 & -1 & -1 & -1 \\ -3 & -1 & 0 & -1 & 3 \\ 1 & -1 & -1 & 0 & -1 \\ -3 & -1 & 3 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 8 \\ 2 \\ -8 \\ 2 \\ -8 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

Finally the system arrives to a locally stable point limit when $x^1(2) = x^1(1)$, therefore the iteration process ends with a bad recalled pattern.

The recalling phase for the non iterative Hopfiel model is computed with the matrix **M**. According to definition 1 and 2, the thresholded vector u_i and the corrective vector are calculated using (10) and (12) as follows:

$$\boldsymbol{u_i^1} = \begin{pmatrix} -6 \\ -3 \\ 0 \\ -3 \\ 0 \end{pmatrix} \qquad \qquad \boldsymbol{z_i^1} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$

Using definition 3, the adjusted vector y_i is obtained with (12) expression as follows:

$$\boldsymbol{y_i^1} = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \begin{matrix} = \\ \neq \\ = \\ = \\ = \end{matrix} \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

In order to estimate the performance of the non iterative Hopfield model with real data information, a database of 20 binary patterns was made to obtain a sensibility analysis. Different letters images were used as fundamental patterns (Fig. 1).

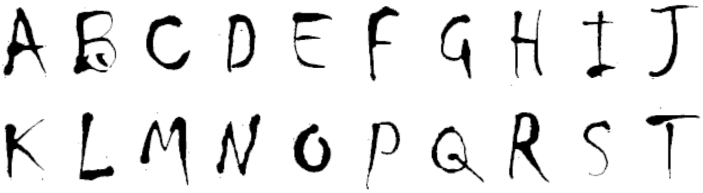


Fig. 1. Fundamental set of letters used in the iterative and non iterative Hopfield model analysis

Each of the fundamental patters produces a 60x90 bits vector. Accordingly, each pattern association has a 5400 bits vector, thus a 20 by 5400 matrix is obtained after the learning phase is completed.

A comparison between recalling algorithms is computed in order to determine their performances. Same process was done using the methodologies developed by [4]. The recalling results are shown in Table 1. An interesting conclusion in the recalling criterion that was used along the experimental phase is that, in this case, a perfect recalling was achieved with the 5400 bits vectors using the proposed method.

Table 1. Classification results for the Hopfield associative memory (HAM), proposed model by [4] (algorithm 1), and the non iterative Hopfield associative memory (NHAM)

	HAM		Algorithm 1		**NHAM**	
Patterns number	Recalled patterns	Error recalling	Recalled patterns	Error recalling	**Recalled patterns**	**Error recalling**
5	5	0	5	0	**5**	**0**
10	9	1	10	0	**10**	**0**
15	13	2	14	1	**15**	**0**
20	16	4	17	3	**20**	**0**

6 Conclusions

In this paper a new algorithm which is based on the first known model of Iterative Hopfield memory has been introduced. The Hopfield iterations are avoided using a corrective vector which is computed with the Hopfield memory. The algorithm applies the corrective vector over bad recalled class vectors in order to obtain a good class vector. Experimental results have shown that this algorithm is an efficient way to improve classifier accuracy.

Acknowledgments. The authors of the present paper would like to thank the following institutions for their support to develop this work: National Polytechnic Institute, Mexico and, CONACyT.

References

1. Acevedo, M.: Memorias asociativas bidireccionales alfa – beta. Ph.D. Thesis, Centre of Computer Research, Instituto Politécnico National. México (2006)
2. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79, 2554–2558 (1982)
3. Abu, M., Jacques, J.: Information capacity of the Hopfield model. IEEE Transactions on Information Theory IT-31(4), 461–464 (1985)
4. Catalán, E., Yáñez, C.: Non iterative Hopfield model. In: Proceedings of the Electronics, Robotics and Automotive Mechanics Conference, CERMA 2006 (2006)
5. Román, G., López, I., Yáñez, M.: A new classifier based on Associative Memories. In: Proc. 15th International Conference on Computing CIC 2006, pp. 55–59. IEEE Computer Society, Los Alamitos (2006)
6. Palm, G., Schwenker, F., Sommer, F.T., Strey, A.: Neural associative memories. In: Krikelis, A., Weems, C.C. (eds.) Associative Processing and Processors, pp. 307–326. IEEE Computer Society, Los Alamitos (1997)
7. Yañez, C.: Memorias Asociativas basadas en relaciones de orden y operadores binarios. Ph.D. Thesis. Center of Computer Research, Instituto Politécnico National. México (2002)
8. Anderson, J., Bower, G.: Memoria Asociativa, México, Limusa (1977)
9. Flores, R., Yañez, C.: Memorias asociativas alfa beta basadas en código Johnson Möbius modificado. Ms.C. Thesis. Center of Computer Research, Instituto Politécnico National. México (2006)

Author Index