Ignacio Rojas
Gonzalo Joya
Andreu Catala (Eds.)

Advances in Computational Intelligence

LNCS 9094

13th International Work-Conference on Artificial Neural Networks, IWANN 2015
Palma de Mallorca, Spain, June 10–12, 2015
Proceedings, Part I

Springer

Lecture Notes in Computer Science 9094

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

Editorial Board

More information about this series at http://www.springer.com/series/7407

Ignacio Rojas · Gonzalo Joya
Andreu Catala (Eds.)

Advances in Computational Intelligence

13th International Work-Conference on Artificial Neural Networks, IWANN 2015 Palma de Mallorca, Spain, June 10–12, 2015 Proceedings, Part I

Springer

Editors
Ignacio Rojas
University of Granada
Granada
Spain

Andreu Catala
Polytechnic University of Catalonia
Vilanova i la Geltrú
Spain

Gonzalo Joya
University of Malaga
Malaga
Spain

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-19257-4 ISBN 978-3-319-19258-1 (eBook)
DOI 10.1007/978-3-319-19258-1

Library of Congress Control Number: 2015939427

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

We are proud to present the set of final accepted papers for the 13th edition of the IWANN conference "International Work-Conference on Artificial Neural Networks" held in Palma de Mallorca (Spain) during June 10–12, 2015.

IWANN is a biennial conference that seeks to provide a discussion forum for scientists, engineers, educators, and students about the latest ideas and realizations in the foundations, theory, models, and applications of hybrid systems inspired on nature (neural networks, fuzzy logic, and evolutionary systems) as well as in emerging areas related to the above items. As in previous editions of IWANN, it also aims to create a friendly environment that could lead to the establishment of scientific collaborations and exchanges among attendees. The proceedings will include all the presented communications to the conference. It has also foreseen the publication of an extended version of selected papers in a special issue of several specialized journals (such as Neurocomputing, Soft Computing, and Neural Processing Letters).

Since the first edition in Granada (LNCS 540, 1991), the conference has evolved and matured. The list of topics in the successive Call for Papers has also evolved, resulting in the following list for the present edition:

1. **Mathematical and theoretical methods in computational intelligence**. Mathematics for neural networks. RBF structures. Self-organizing networks and methods. Support vector machines and kernel methods. Fuzzy logic. Evolutionary and genetic algorithms.
2. **Neurocomputational formulations**. Single-neuron modelling. Perceptual modelling. System-level neural modelling. Spiking neurons. Models of biological learning.
3. **Learning and adaptation**. Adaptive systems. Imitation learning. Reconfigurable systems. Supervised, non-supervised, reinforcement, and statistical algorithms.
4. **Emulation of cognitive functions**. Decision Making. Multi-agent systems. Sensor mesh. Natural language. Pattern recognition. Perceptual and motor functions (visual, auditory, tactile, virtual reality, etc.). Robotics. Planning motor control.
5. **Bio-inspired systems and neuro-engineering**. Embedded intelligent systems. Evolvable computing. Evolving hardware. Microelectronics for neural, fuzzy, and bioinspired systems. Neural prostheses. Retinomorphic systems. Brain–computer interfaces (BCI). Nanosystems. Nanocognitive systems.
6. **Advanced topics in computational intelligence**. Intelligent networks. Knowledge-intensive problem-solving techniques. Multi-sensor data fusion using computational intelligence. Search and meta-heuristics. Soft Computing. Neuro-fuzzy systems. Neuro-evolutionary systems. Neuro-swarm. Hybridization with novel computing paradigms.
7. **Applications**. Expert Systems. Image and Signal Processing. Ambient intelligence. Biomimetic applications. System identification, process control, and manufacturing. Computational Biology and Bioinformatics. Parallel and Distributed Computing.

Human–Computer Interaction, Internet Modeling, Communication and Networking. Intelligent Systems in Education. Human–Robot Interaction. Multi-Agent Systems. Time series analysis and prediction. Data mining and knowledge discovery.

At the end of the submission process, and after a careful peer-review and evaluation process (each submission was reviewed by at least 2, and on the average 2.7, Program Committee members or additional reviewers), 100 papers were accepted for oral or poster presentation, according to the recommendations of reviewers and the authors' preferences.

It is important to note, that for the sake of consistency and readability of the book, the presented papers are not organized as they were presented in the IWANN 2015 sessions, but classified under 14 chapters. The organization of the papers is in two volumes arranged basically following the topics list included in the Call for Papers. The first volume (LNCS 9094), entitled "IWANN 2015. Advances on Computational Intelligence. Part I" is divided into eight main parts and includes the contributions on:

1. Computing Languages with Bio-Inspired Devices and Multi-Agent Systems (Special Session, organized by: M. Dolores Jiménez-López and Alfonso Ortega de la Puente)
2. Brain-Computer Interfaces: Applications and Tele-services (Special Session, organized by: Ricardo Ron Angevin and Miguel Angel Lopez)
3. Multi-Robot Systems: Applications and Theory (MRSAT) (Special Session, organized by: José Guerrero and Óscar Valero)
4. Video and Image Processing (Special Session, organized by: Enrique Domínguez and Jose Garcia)
5. Transfer Learning (Special Session, organized by: Luis M. Silva and Jorge M. Santos)
6. Structures, algorithms, and methods in artificial intelligence (Special Session, organized by: Daniela Danciu and Vladimir Răsvan)
7. Interactive and Cognitive Environments (Special Session, organized by: Wei Chen and Albert Samá)
8. Mathematical and theoretical methods in Fuzzy Systems

In the second volume (LNCS 9095), entitled "IWANN 2015. Advances on Computational Intelligence. Part II" is divided into six main parts and includes the contributions on:

1. Pattern Recognition
2. Embedded intelligent systems
3. Expert Systems
4. Advances in Computational Intelligence
5. Applications of Computational Intelligence
6. Invited Talks to IWANN 2015

In this edition of IWANN 2015, the plenary talks were given by Prof. Cristina Urdiales (The shared control paradigm for assistive and rehabilitation robots), Prof. Dan Ciresan (Deep Neural Networks for Visual Pattern Recognition), and finally by Prof. Andrea Cavallaro.

The 13th edition of the IWANN conference was organized by the University of Granada, University of Málaga, Polytechnic University of Catalonia, and University of the Balearic Islands, together with the Spanish Chapter of the IEEE Computational Intelligence Society. We wish to thank the University of the Balearic Islands for their support and grants.

We would also like to express our gratitude to the members of the different committees for their support, collaboration, and good work. We specially thank the Local Committe, Program Committee, the Reviewers, Invited Speaker, and Special Session Organizers. Finally, we want to thank Springer, and especially Alfred Hoffman and Anna Kramer for their continuous support and cooperation.

June 2015

Ignacio Rojas
Gonzalo Joya
Andreu Catala

Organization

Program Committee

Leopoldo Acosta	University of La Laguna, Spain
Vanessa Aguiar-Pulido	RNASA-IMEDIR, University of A Coruña, Spain
Arnulfo Alanis Garza	Instituto Tecnológico de Tijuana, Mexico
Ali Fuat Alkaya	Marmara University, Turkey
Amparo Alonso-Betanzos	University of A Coruña, Spain
Juan Antonio Alvarez-García	University of Seville, Spain
Jhon Edgar Amaya	University of Tachira (UNET), Venezuela
Gabriela Andrejkova	Pavol Jozef Šafarik University, Slovak Republic
Cesar Andres	Universidad Complutense de Madrid, Spain
Miguel Ángel López	University of Cádiz, Spain
Anastassia Angelopoulou	University of Westminster, UK
Plamen Angelov	Lancaster University, UK
Davide Anguita	University of Genova, Italy
Cecilio Angulo	Universitat Politècnica de Catalunya, Spain
Javier Antich	Universitat de les Illes Balears, Spain
Angelo Arleo	CNRS - University Pierre and Marie Curie Paris VI, France
Corneliu Arsene	SC IPA SA, Romania
Miguel Atencia	University of Málaga, Spain
Jorge Azorín-López	University of Alicante, Spain
Davide Bacciu	University of Pisa, Italy
Javier Bajo	Universidad Politécnica de Madrid, Spain
Juan Pedro Bandera Rubio	ISIS Group, University of Málaga, Spain
Cristian Barrué	Universitat Politècnica de Catalunya, Spain
Andrzej Bartoszewicz	Technical University of Lodz, Poland
Bruno Baruque	University of Burgos, Spain
David Becerra Alonso	University of the West of Scotland, UK
Lluís Belanche	Universitat Politècnica de Catalunya, Spain
Sergio Bermejo	Universitat Politècnica de Catalunya, Spain
Francesc Bonin	Universitat de les Illes Balears, Spain
Francisco Bonnín Pascual	Universitat de les Illes Balears, Spain
Julio Brito	University of La Laguna, Spain
Antoni Burguera	Universitat de les Illes Balears, Spain
Joan Cabestany	Universitat Politècnica de Catalunya, Spain
Inma P. Cabrera	University of Málaga, Spain
Tomasa Calvo	University of Alcalá, Spain

Jose Luis Calvo Rolle	University of A Coruña, Spain
Francesco Camastra	University of Naples Parthenope, Italy
Carlos Carrascosa	GTI-IA DSIC Universidad Politecnica de Valencia, Spain
Luis Castedo	Universidad de A Coruña, Spain
Pedro Castillo	University of Granada, Spain
Andreu Catala	Universitat Politècnica de Catalunya, Spain
Ana Cavalli	GET/INT, France
Miguel Cazorla	University of Alicante, Spain
Wei Chen	Eindhoven University of Technology, The Netherlands
Jesus Cid-Sueiro	Universidad Carlos III de Madrid, Spain
Maximo Cobos	Universidad de Valencia, Spain
Valentina Colla	Scuola Superiore Sant'Anna, Italy
Pablo Cordero	Universidad de Málaga, Spain
Francesco Corona	Aalto University, Finland
Ulises Cortes	Universitat Politècnica de Catalunya, Spain
Marie Cottrell	SAMM Université Paris 1 Panthéon-Sorbonne, France
Raúl Cruz-Barbosa	Universidad Tecnológica de la Mixteca, Mexico
Manuel Cruz-Ramírez	University of Córdoba, Spain
Erzsébet Csuhaj-Varjú	Eötvös Loránd University, Hungary
Daniela Danciu	University of Craiova, Romania
Suash Deb	C.V. Raman College of Engineering, India
Angel Pascual Del Pobil	UNIVERSITAT JAUME I, Spain
Enrique Domínguez	University of Málaga, Spain
Julian Dorado	Universidade da Coruña, Spain
Abrahan Duarte	Universidad Rey Juan Carlos, Spain
Richard Duro	Universidad de A Coruña, Spain
Gregorio Díaz	University of Castilla - La Mancha, Spain
Emil Eirola	Aalto University, Finland
Patrik Eklund	Umeå University, Sweden
Javier Fernández de Cañete	University of Málaga, Spain
Francisco Fernandez De Vega	Grupo de Evolucion Artificial, Universidad de Extremadura, Spain
Alberto Fernandez Gil	CETINIA, Universidad Rey Juan Carlos, Spain
Enrique Fernandez-Blanco	University of A Coruña, Spain
Manuel Fernández Carmona	Universidad de Málaga, Spain
Antonio J. Fernández Leiva	Universidad de Málaga, Spain
Francisco Fernández Navarro	University of Córdoba, Spain

Carlos Fernández-Lozano	Universidade da Coruña, Spain
Jose Manuel Ferrandez	Universidad Politécnica de Cartagena, Spain
Ricardo Ferreira	Nove de Julho University, Brazil
Aníbal R. Figueiras-Vidal	Universidad Carlos III de Madrid, Spain
Oscar Fontenla-Romero	University of A Coruña, Spain
Colin Fyfe	University of the West of Scotland, UK
Emilio Garcia	Universitat de les Illes Balears, Spain
Rodolfo García-Bermudez	Universidad Laica Eloy Alfaro de Manabí, Ecuador
Carlos Garcia Puntonet	University of Granada, Spain
Juan M Garcia-Gomez	Universidad Politecnica de Valencia, Spain
Francisco Garcia-Lagos	Universidad de Málaga, Spain
Jose Garcia-Rodriguez	University of Alicante, Spain
Patricio García Báez	Universidad de La Laguna, Spain
Pablo García Sánchez	University of Granada, Spain
Maribel García-Arenas	Universidad de Granada, Spain
Patrick Garda	Université Pierre et Marie Curie - Paris 6, France
Peter Gloesekoetter	Münster University of Applied Sciences, Germany
Juan Gomez Romero	Universidad Carlos III de Madrid, Spain
Juan Gorriz	University of Granada, Spain
Karl Goser	Technical University Dortmund, Germany
Bernard Gosselin	University of Mons, Belgium
Manuel Grana	University of the Basque Country, Spain
Bertha Guijarro-Berdiñas	University of A Coruña, Spain
Nicolás Guil Mata	University of Málaga, Spain
Alberto Guillén	University of Granada, Spain
Francisco Herrera	University of Granada, Spain
Álvaro Herrero	University of Burgos, Spain
Cesar Hervas	University of Cordoba, Spain
Tom Heskes	Radboud University Nijmegen, The Netherlands
Wei-Chiang Hong	Oriental Institute of Technology, India
José M. Jerez	Universidad de Málaga, Spain
M. Dolores Jimenez-Lopez	Rovira i Virgili University, Spain
Juan Luis Jiménez Laredo	University of Granada, Spain
Gonzalo Joya	University of Málaga, Spain
Vicente Julian	Universidad Politécnica de Valencia, Spain
Fernando L. Pelayo	University of Castilla - La Mancha, Spain
Alberto Labarga	University of Granada, Spain
Raul Lara Cabrera	University of Málaga, Spain

Nuno Lau	Universidade de Aveiro, Portugal
Amaury Lendasse	The University of Iowa, USA
Miguel Lopez	University of Granada, Spain
Otoniel López Granado	Miguel Hernandez University, Spain
Rafael Marcos Luque Baena	University of Málaga, Spain
Ezequiel López-Rubio	University of Málaga, Spain
Kurosh Madani	LISSI/Université PARIS-EST Creteil, France
Mario Martin	Universitat Politècnica de Catalunya, Spain
Bonifacio Martin Del Brio	University of Zaragoza, Spain
Jose D. Martin-Guerrero	University of Valencia, Spain
Luis Martí	Universidad Carlos III de Madrid, Spain
Francisco Martínez Estudillo	ETEA, Spain
José Luis Martínez Martínez	University of Castilla-La Mancha, Spain
José Fco. Martínez-Trinidad	INAOE, Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico
Miquel Massot	University of the Balearic Islands, Spain
Francesco Masulli	University of Genoa, Italy
Montserrat Mateos	Universidad Pontificia de Salamanca, Spain
Jesús Medina-Moreno	University of Cádiz, Morocco
Maria Belen Melian Batista	University of La Laguna, Spain
Mercedes Merayo	Universidad Complutense de Madrid, Spain
Gustavo Meschino	Universidad Nacional de Mar del Plata, Spain
Margaret Miro	University of the Balearic Islands, Spain
Jose M. Molina	Universidad Carlos III de Madrid, Spain
Augusto Montisci	University of Cagliari, Italy
Antonio Mora	University of Granada, Spain
Angel Mora Bonilla	University of Málaga, Spain
Claudio Moraga	European Centre for Soft Computing, Spain
Ginés Moreno	University of Castilla-La Mancha, Spain
Jose Andres Moreno	University of La Laguna, Spain
Juan Moreno Garcia	Universidad de Castilla-La Mancha, Spain
J. Marcos Moreno Vega	University of La Laguna, Spain
Susana Muñoz Hernández	Technical University of Madrid, Spain
Pep Lluís Negre Carrasco	University of the Balearic Islands, Spain
Alberto Núñez	Universidad de Castilla La Mancha, Spain
Manuel Ojeda-Aciego	University of Málaga, Spain
Sorin Olaru	"SUPELEC" École **Su**périeur d'**Élec**tricité, France
Iván Olier	The University of Manchester, UK
Madalina Olteanu	SAMM, Université Paris 1, France
Julio Ortega	Universidad de Granada, Spain
Alfonso Ortega de La Puente	Universidad Autónoma de Madrid, Spain
Alberto Ortiz	University of the Balearic Islands, Spain
Emilio Ortiz-García	Universidad de Alcalá, Spain

Osvaldo Pacheco	Universidade de Aveiro, Portugal
Esteban José Palomo	University of Málaga, Spain
Diego Pardo	Barcelona Tech, Spain
Miguel Angel Patricio	Universidad Carlos III de Madrid, Spain
Alejandro Pazos Sierra	University of A Coruña, Spain
Jose Manuel Perez Lorenzo	Universidad de Jaén, Spain
Vincenzo Piuri	University of Milan, Italy
Héctor Pomares	University of Granada, Spain
Alberto Prieto	Universidad de Granada, Spain
Alexandra Psarrou	University of Westminster, UK
Francisco A. Pujol	University of Alicante, Spain
Pablo Rabanal	Universidad Complutense de Madrid, Spain
Juan Rabuñal	University of A Coruña, Spain
Vladimir Răsvan	Universitatea din Craiova, Romania
Ismael Rodríguez	Universidad Complutense de Madrid, Spain
Juan A. Rodriguez	Universidad de Málaga, Spain
Sara Rodríguez	University of Salamanca, Spain
Fernando Rojas	University of Granada, Spain
Ignacio Rojas	University of Granada, Spain
Samuel Romero-Garcia	University of Granada, Spain
Ricardo Ron-Angevin	University of Málaga, Spain
Eduardo Ros	University of Granada, Spain
Francesc Rossello	University of the Balearic Islands, Spain
Fabrice Rossi	SAMM, Université Paris 1, France
Fernando Rubio	Universidad Complutense de Madrid, Spain
Ulrich Rueckert	University of Paderborn, Germany
Addisson Salazar	Universidad Politécnica de Valencia, Spain
Sancho Salcedo-Sanz	Universidad de Alcalá, Spain
Albert Samà	Universitat Politècnica de Catalunya, Spain
Francisco Sandoval	Universidad de Málaga, Spain
José Santos	University of A Coruña, Spain
Jose A. Seoane	University of Bristol, UK
Eduardo Serrano	Universidad Autónoma de Madrid, Spain
Luís Silva	University of Aveiro, Portugal
Olli Simula	Helsinki University of Technology, Finland
Jordi Solé-Casals	Universitat de Vic, Spain
Carmen Paz Suárez Araujo	Universidad de las Palmas de Gran Canaria, Spain
Peter Szolgay	Pázmány Péter Catholic University, Hungary
Javier Sánchez-Monedero	University of Córdoba, Spain
Ricardo Tellez	Pal Robotics, Spain
Ana Maria Tome	Universidade de Aveiro, Portugal
Carme Torras	IRI (CSIC-UPC), Spain
Joan Torrens	University of the Balearic Islands, Spain

Claude Touzet	University of Provence, France
Olga Valenzuela	University of Granada, Spain
Óscar Valero	University of the Balearic Islands, Spain
Miguel Ángel Veganzones	Universidad del País Vasco (UPV/EHU), Spain
Francisco Velasco-Álvarez	Universidad de Málaga, Spain
Sergio Velastin	Kingston University, UK
Marley Vellasco	PUC-Rio, Brazil
Alfredo Vellido	Universitat Politècnica de Catalunya, Spain
Francisco J. Veredas	Universidad de Málaga, Spain
Michel Verleysen	Université catholique de Louvain, Belgium
Changjiu Zhou	Singapore Polytechnic, Singapore
Ahmed Zobaa	University of Exeter, UK

Additional Reviewers

Azorín-López, Jorge	Navarro-Ortiz, Jorge
Cortes, Ulises	Ortiz, Alberto
De La Cruz, Marina	Palomo, Esteban José
Fernandez-Blanco, Enrique	Peters, Peter
Garcia-Fidalgo, Emilio	Rodriguez, Juan A.
Georgieva, Petia	Rodriguez-Benitez, Luis
Luque-Baena, Rafael M.	Sánchez-Morillo, Daniel
López-Rubio, Ezequiel	Volosyak, Ivan
Martinez-Gomez, Jesus	Wang, Qi
Moreno Garcia, Juan	Wetzels, Mart

Contents – Part I

Transfer Learning

Contents – Part II

Pattern Recognition

Computing Languages with Bio-Inspired Devices and Multi-Agent Systems

A Grammatical Inference Model for Measuring Language Complexity

Leonor Becerra-Bonache[1] and M. Dolores Jiménez-López[2(✉)]

[1] Laboratoire Hubert Curien, Jean Monnet University, 18 rue Benoit Lauras, 42100 Saint-Etienne, France
leonor.becerra@univ-st-etienne.fr

[2] Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Av. Catalunya, 35, 43002 Tarragona, Spain
mariadolores.jimenez@urv.cat

Abstract. The 21st century has re-opened the interest of Linguistics on the complexity of natural languages. The equi-complexity dogma –the idea that all languages must be equally complex– has been challenged by a number of researchers that claim that indeed natural languages differ in complexity. In the last fifteen years, challengers of the equi-complexity dogma have proposed many complexity measures that depend on their way of defining complexity. In this paper, we propose a grammatical inference model to measure the relative complexity of languages. The computational tool we introduce is the result of an interdisciplinary study inspired in the process of natural language acquisition.

Keywords: Linguistic complexity · Relative complexity · Grammatical inference · Machine Learning

1 Introduction

This paper focuses on one of the basic assumptions of linguistics: the *linguistic equi-complexity dogma*. When asked if all languages are equally complex, twentieth century most theoretical linguists have answered with the principle of invariance in language complexity, defending the equilibrium hypothesis, which states that the total complexity of a language is fixed because sub-complexities in linguistic sub-systems trade off. This idea of equi-complexity, seen for decades as an unquestioned truism of Linguistics, has begun to be explicitly questioned in recent years [6,15,16,20,21,23].

Many models have been proposed to confirm or refute the hypothesis of linguistic equi-complexity. The tools, criteria and measures to quantify the level of complexity of languages vary and depend on the specific research interests and on the definition of complexity adopted. Currently, there is no clear solution to quantify the complexity of languages and each of the proposed models has advantages and disadvantages.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 3–17, 2015.
DOI: 10.1007/978-3-319-19258-1_1

The main objective of this paper is to demonstrate the differences in the levels of complexity of natural languages by providing an objective and meaningful method to calculate linguistic complexity. To achieve this goal, we propose an interdisciplinary solution that uses a solidly defined computational model to quantify the cost/difficulty in the process of acquisition of different languages (relative complexity), showing that it is not identical in all cases. The computational model we propose is inspired by the process of language acquisition and is included in the field of *Grammatical Inference*, a *Machine Learning* subdiscipline.

The paper is organized as follows. Firstly, we present an overview of research on linguistic complexity by summarizing the main types of complexity definitions and the variety of metrics that have been proposed to quantify natural language complexity. Secondly, we briefly introduce the field of grammatical inference. Thirdly, we present a grammatical inference algorithm that calculates the complexity of natural languages in relative terms (cost/ difficulty of processing). Finally, we conclude with some remarks and possible directions for future work.

2 Linguistic Complexity

Are all languages equally complex? Does it make sense to compare the complexity of languages? Can languages differ in complexity? If we analyze the answers to these questions we can find two different types of responses: those who have agreed that all languages are equal in terms of complexity and those who consider that it is relevant to talk about different levels of linguistic complexity.

The first type of answer predominates in the 20th century Linguistics. It has been defended for a long time that the linguistic complexity is invariant and that languages are not measurable in terms of complexity. Those ideas have been dubbed the *ALEC statement* ('All Language are Equally Complex') [8] or the *linguistic equi-complexity dogma* [15].

Three different structural interpretations of the thesis that ?all languages are equal? can be found in the literature [27]:

- the *total grammatical complexity*, where language complexity is equal for all languages, and interdomain complexity balances;
- the *minimum argument*, where there is a minimum threshold for language structure, and everything over that threshold is equal. Languages can differ in complexity; all that is important is that they meet the standard;
- the *maximum argument*, where there is a maximum threshold for language complexity; there is an upper limit above which no language can be more complex.

Of those three structural interpretations, one of them is discussed far more than the other two: it is the *total grammatical complexity*. In fact, the general idea behind the ALEC statement or the equi-complexity dogma 'is that the total complexity of a language is fixed because sub-complexities in linguistic subsystems trade off' [14]. This interpretation is clearly related to Hockett [12]

who, with the following assertion, first proposed total grammatical complexity as an interpretation of the thesis that all language are equal:

> Objective measurement is difficult, but impressionistically it would seem that the total grammatical complexity of any language, counting both morphology and syntax, is about the same as that of any other. This is not surprising, since all languages have about equally complex jobs to do, and what is not done morphologically has to be done syntactically. Fox, with a more complex morphology than English, thus ought to have a somewhat simpler syntax; and this is the case [12].

The questions that arise in front of the ALEC statement are obvious: If languages differ in the complexity of individual subsystems (this is supported by the equi-complexity axiom), why is the total complexity always the same? What mechanism stops the complexity in one area when the complexity has increased in another one? Why complexity in one sub-system should be compensated by simplicity in another? What is the factor responsible for the equi-complexity? Those questions have opened again the debate on linguistic complexity. The beginning of the 21st century have witnessed an important number of challengers of the equi-complexity dogma that have tried to show that high levels of complexity in some domain of a given language do not necessarily entail low local complexity in some other domain of the same language.

The article by McWhorter [16] in the special volume of the journal Linguistic Typology is considered one of the primers to re-open the interest in the study of linguistic complexity. From this pioneering work, there have been many seminars, conference, articles, monographs [6,15] and collective volumes [20,21,23] that have dealt with the topic of linguistic complexity and that have challenged the equi-complexity dogma. In those works, the notion of complexity have been understood in many different ways and there have been many measurement proposals. In what follows, we will try to briefly summarize those different approaches to the concept of complexity

2.1 Linguistic Complexity Typology

There is agreement in the literature about the difficulty of defining complexity. This difficulty has had as a consequence an important number of different types of definitions of the notion. In fact, a unified definition of complexity does not exist. Instead, in the literature, we can find a variety of approaches that has led to a linguistic complexity taxonomy. In what follows, we will try to summarize some of the most common types of linguistic complexity.

Fist of all, it is important to distinguish between *absolute* complexity and *relative* complexity [17–19]:

- The *absolute complexity* –or *objective complexity*– approach defines complexity as an objective property of the system. It is measured in terms of the number of parts of the system, the interrelations between the parts or the

length of the description of the phenomenon. It is a usual complexity notion in cross-linguistic typology studies [6,16].

– The *relative complexity* –or *agent-related complexity*– approach takes into account the users of language. In this category, complexity is identified with difficulty/cost of processing, learning or acquisition. In general, it is accepted that what is costly or difficult to language users is complex. This type of approach is very common in the fields of sociolinguistics and psycholinguistics [15].

Some authors [6] prefer to reserve the term *complexity* for absolute complexity and to use other terms such as *cost, difficulty* or *demandingness* to denote relative complexity.

Another common dichotomy in the literature is the one that distinguish *global* complexity from *local* complexity [18]:

– *Global complexity* is understood as complexity of a language as such. It is about the overall complexity of an entity. It calculates the total complexity of the linguistic system.

– *Local complexity* is understood as domain-specific complexity. This type of complexity is about some part of an entity. It analyzes the complexity of particular sub-domains of the language. When concerned with measures of local complexity, it is very frequent to focus on one of the following subdomains [14]: 1) *phonological complexity*, interested in measuring the size of phoneme inventory, the suprasegmental phonology or the phonotactic restrictions; 2) *morphological complexity*, where it is specially important to consider inflectional morphology; 3) *syntactic complexity*, where the number of rules or clausal embedding is measured; 4) *semantic* and *lexical complexity*, where homonymous and polysemous words or lexical variation are important; and 5) *pragmatic complexity*, where inference is a central topic.

It is also very frequent in the linguistic complexity bibliography to distinguish between *system* complexity and *structural* complexity [6]:

– *System complexity* considers the properties of the language and calculates the content of the speaker's competence. It can be understood as a measure of the content that language learners have to master in order to be proficient in language.

– *Structural complexity* calculates the quantity of structure of a linguistic object, analyzes the structure of the expressions. It is about the complexity of expressions at some level of description.

More specific complexity notions of complexity –*absolute-quantitative complexity, redundancy-induced complexity, irregularity-induced complexity, L2 acquisition complexity*– are discussed in [14].

Pallotti [22] summarizes the different notions of 'complexity' in the field of linguistics by referring to three main meanings:

- *Structural complexity*, a formal property of texts and linguistic systems having to do with the number of their elements and their relational patterns.
- *Cognitive complexity*, having to do with the processing costs associated with linguistic structures.
- *Developmental complexity*, the order in which linguistic structures emerge and are mastered in second (and, possibly, first) language acquisition.

2.2 Metrics of Linguistic Complexity

There is no conventionally agreed metric for measuring the complexity of natural languages. In fact, in the field of linguistic complexity many *ad hoc* complexity measures have been proposed. Edmonds [9], for example, identifies forty-eight different metrics used in natural and social sciences. The measures proposed are varied and could be grouped into two blocs:

- *Measures of absolute complexity.* The number of categories or rules, length of the description, ambiguity, redundancy, etc. [18].
- *Measures of relative complexity.* When the relative approach of complexity is adopted the problem that has to be faced is the answer to the question: Difficult/costly to whom? This means that it is necessary to determine what kind of task –learning, acquisition, processing– must be considered and, of course, what type of user must be taken into account –speaker, listener, child, adult–. The problem, here, is to choose a task and a type of user representative for the definition of complexity and to properly motivate this choice. The complexity of L2 learning [15] or the complexity of processing [11] are examples of measures proposed in this field.

The magnitude of the problem requires an interdisciplinary solution. That is why, some researchers have attempted to apply the concept of complexity used in other disciplines in order to find useful tools to calculate linguistic complexity. Information theory, for example, offers two formalisms that might be appropriate for measuring linguistic complexity:

- *Shannon information entropy* that captures the average number of bits of information necessary to specify the state of a random variable or system described by a probability model [4].
- *Kolmogorov complexity* that measures the informativeness of a string as the length of the algorithm required to describe than string [13]. This measure can be applied to measuring language complexity in such a way that the longer the description of a linguistic structure, the more complex it is [6,13, 18].

Other than information theory, computational models [5], or the theory of complex systems [1] are examples of areas that provide measures to quantitatively evaluate linguistic complexity.

The variety of proposals shows that there is not an unanimously accepted solution to quantify the linguistic complexity. Taking into account the necessity of interdisciplinarity, we propose a Machine Learning model, defined in the field of Grammatical Inference, to measure the relative complexity of languages.

3 Grammatical Inference

The field of Machine Learning is focused on the development of techniques that allow computers to learn. *Grammatical Inference* (GI) is a subfield of Machine Learning that deals with the learning of *formal languages*.

Formal languages are defined as follows. Given an alphabet Σ, a finite sequence of symbols chosen from Σ is called a *string*. The set of all strings over the alphabet Σ is denoted by Σ^*. Each subset of Σ^* is called a *language* over the alphabet Σ. A *grammar* is a finite set of rules that generate a language.

Chomsky proposed in the 50's a hierarchical classification of formal grammars, the so-called *Chomsky hierarchy*. According to the form of their rules, Chomsky grammars are classified as: *Regular, Context-Free, Context-Sensitive* and *Recursively Enumerable*. The family of languages generated by these four types of grammars are denoted by REG, CF, CS and RE, respectively. The following strict inclusions hold: $REG \subset CF \subset CS \subset RE$.

GI refers to the process of learning grammars and languages from data. E.M. Gold gave the initial theoretical foundations of this field in 1967 [10]. His goal was to construct a language learning model in order to investigate from a theoretical point of view how a machine can achieve the ability of speaking a language. Since Gold's work, GI studies have been specially focused on obtaining formal results, i.e. on providing formal descriptions of the target languages (i.e., languages to be learned), formal proofs about the efficiency of learning algorithms, etc. The classes of languages that have mainly been studied are REG and CF. However, we can also find some practical approaches, where the goal of the researchers is to provide empirical systems that deal with natural language data, rather than proving the learnability of grammars. For detailed information about the results that have been obtained in GI, see [7].

In a GI problem, we have a *teacher* that provides information about the *target language* to the *learner* (or learning algorithm), and the learner must identify the underlying language from the data it receives. For example, imagine that the target language is $L = ab^+$. If the teacher provides to the learner strings that belong to the language, such as $ab, abb, abbb$, the learner should infer that the target language is $L = ab^+$.

According to E.M. Gold [10], a *language learnability model* requires: 1) a definition of learnability; 2) a method to provide information to the learning algorithm; and 3) a mathematical object to describe the languages (which could correspond, for example, to a grammar).

Several formal models have been proposed in GI. The most studied ones are: *Identification in the limit* [10], *Active learning* [2] and *PAC learning* [26]. These three models are focused on syntax learning.

3.1 Syntactic Models of Language Learning

The model of *Identification in the limit* was proposed by E.M. Gold in [10]. In this model, the learning algorithm takes as input a finite sequence of examples and gives as output a hypothesis of the target language. More concretely, at each time t, the learner receives a unit of information concerning the target language, and the learner makes a guess of the target language based on the information received through time t. Hence, if the new example received is not consistent with the current learner's hypothesis, the learner has to change its hypothesis. The learner identifies the target language *in the limit* if, after some finite time, makes a correct hypothesis and does not change it.

Note that, in this model, the learner does not necessarily know when its hypothesis is correct. Gold's justification for studying identifiability in the limit was the following:

> A person does not know when he is speaking a language correctly; there is always the possibility that he will find that his grammar contains an error. But we can guarantee that a child will eventually learn a natural language, even if it will not know when it is correct [10].

Two methods of information presentations are possible in this learning paradigm:

- *Text*: a *text* for a language L is a sequence of strings $x_1, x_2, \ldots$ from L, such that every string of L occurs at least once in the text. In this case, we say that the learner learns from only *positive data*. For example, imagine that the target language is $L = \{ab^+\}$. An example of a text presentation would be: *abb*, *abbbbb*, ...
- *Informant*: an *informant* for a language L is a sequence of pairs (x_1, b_1), (x_2, b_2), ... in $L \times \mathbb{B}$ (where $\mathbb{B}$ is the set of Booleans), such that $b_i = true \Leftrightarrow x_i \in L$. In this case, we say that the learner learns from *positive* and *negative data*. Following the previous example, an informant presentation would be: $(\lambda, false)$, $(abbb, true)$, ...

The model known as *Active learning* was introduced by D. Angluin in [2]. In this model, the learner is allowed to *interact* with the teacher by asking specific questions about the language. Her motivation to introduce this learning framework was the following:

> Imagine a human expert in some domain, for example, cancer diagnosis, attempting to communicate the method he or she uses in that domain to an expert system. Specific positive and negative examples will form an important component of the communication, in addition to advice about general rules, explanations of significant and irrelevant features, justifications of lines of reasoning, clarifications of exceptions, and so on. Moreover, the examples given are likely to be chosen so that they are

> "central" or "crucial" rather than random or arbitrary, in an attempt to speed convergence of the system to a correct hypothesis. [...] Studies of learning general rules from examples have generally assumed a source of examples that is arbitrary or random. The scenario above suggests that it is reasonable to investigate learning methods that assume that the source of examples is "helpful" [2].

In this model, the teacher (also called *oracle*) knows the target language and can correctly answer specific queries asked by the learner. The learner, after asking a finite number of queries, has to return a hypothesis of the language. Exact learning is required, i.e., the learner's hypothesis has to be the correct one.

There exist different types of queries, but the most studied ones are the following:

- *Membership query* (MQ): the learner asks to the teacher if a string x is in the target language L. The teacher answers "yes" if $w \in L$, and "no" otherwise. For example, if the target language is $L = \{ab^+\}$, the learner could ask to the teacher if the following strings are in L: λ, a, b. In this case, the teacher would answer *no* to these three MQ.
- *Equivalence query* (EQ): the learner asks to the teacher if its hypothesis is the correct one. The teacher answers "yes" if the learner's hypothesis is equal to the target language, and if this is not the case, a counterexample is returned to the learner (i.e., a string in the symmetric difference of the learner's hypothesis and the target language).

The *PAC (Probably Approximately Correct) learning model* was introduced by L.G. Valiant in [26]. He proposed a distribution-independent probabilistic model of learning from random examples. In this model, there exists an unknown distribution over the examples, and the learner receives examples sampled under this distribution. It is required to learn under any distribution, but a small error is permitted (i.e., exact learning is not required). A successful learner is one that finds, with high probability, a grammar whose error is small.

For detailed information about results obtained by using these three models, the reader is referred to [7].

4 Grammatical Inference for Linguistic Complexity

In order to measure linguistic complexity with grammatical inference tools, we propose to use a novel model introduced by Angluin and Becerra [3], which takes into account *semantics* during the language learning process.

4.1 The Model

As we have seen in the previous section, the main part of studies in GI reduce the learning problem to the acquisition of the syntax, and omit any semantic

information during the learning process. However, linguists agree that syntactic information is not enough to learn a language and other aspects, like semantics, have to be taken into account.

Angluin and Becerra [3] proposed a GI model that takes into account semantics during the language learning process. In fact, this model, inspired by studies on children's language acquisition, investigates the role of semantics in the process of language learning.

In this model, the teacher and the learner interact in a sequence of situations by producing sentences that denote one object in each situation. These interactions are developed in the following way:

- A situation is randomly generated and it is presented to the teacher and the learner.
- The learner tries to produce a sentence that designs one of the objects in this situation.
- The teacher produces a random sentence that designs one of the objects in this situation.
- The learner analyzes the teacher's sentence and updates its current grammar for the language as appropriate.

Given any situation, the learner's goal is to produce correct sentences that denote one object in this situation. To evaluate the model it was used a simplification of the Feldman task [25] known as the *Miniature Language Acquisition task*. This task consists on learning a sublanguage from sentences-picture pairs that involve geometric figures.

In this model, semantics is formalized by using first order logic. Some of the main formal features of the semantic model are the following. More details can be found in [3].

- **Situation.** A situation is composed of some objects and some of their properties and relations. It is formalized as follows:
  - A finite set of *predicate symbols* P, each of a specific arity (that corresponds to the number of arguments), is used to represent properties of the objects and relations between them. For example, for the property of being "big", the predicate symbol bi_1 is used; for the relation of being 'to the left of', the predicate symbol le_2 is used.
  - A set of *constant symbols* $t_1, t_2, \ldots$ is used to represent different objects.
  - A *ground atom* is an expression formed by applying a predicate symbol to the correct number of constant symbols as arguments. For example, $bl_1(t_1)$.

  Hence, a situation is represented as a finite set of ground atoms over some constants and predicates symbols. For instance, a situation consisting of a big blue triangle to the left of a big green square, is represented by the following set of ground atoms. Both the learner and the teacher have access to this representation of the situation:

$$s_1 = \{bi1(t_1), bl1(t_1), tr1(t_1), le2(t_1, t_2), bi1(t_2), gr1(t_2), sq1(t_2)\}.$$

- **Meaning.** There is a countable number of *variables* $x_1, x_2, \ldots$. A *variable atom* is an expression formed by applying a predicate symbol to the correct number of variables as arguments, for example, $bl_1(x_2)$. A *meaning* is a finite sequence of variable atoms. For example, the meaning of "the blue triangle" is $(bl1(x1), tr1(x1))$. It is assumed that each utterance in the target language is assigned a unique meaning.
- **Utterance.** An utterance is a finite sequence of words over a finite alphabet W of words. For example, *the square* or *square of circle small the blue*. The *target language* is the the set of utterances the teacher may produce in some situation; in our example, this includes *the square* but not *square of circle small the blue*. An utterance is *denoting* in a situation, if it uniquely picks out the objects it refers to in a situation. For example, the utterance *the square* is denoting in the situation s_1 described previously. The goal of the learner is to produce every denoting utterance in any given situation.
- **Meaning Transducers.** The linguistic competence of the teacher is represented by a Finite State Transducer. This transducer is used by the teacher for comprehension and production of utterances. Given a finite set of words W, a finite set of predicates P, and a set of all variable atoms A over P, a meaning transducer M is a 7-tuple such that:

  - Q is a finite set of states;
  - W are input symbols;
  - A are output symbols;
  - $q_0 \in Q$ is an initial state;
  - $F \subseteq Q$ is a finite set of final states;
  - δ is a deterministic transition function mapping $Q \times W$ to Q;
  - γ is an output function mapping $Q \times W$ to $A \cup \{\varepsilon\}$, where ε denotes the empty sequence.

  Initially, the teacher has the meaning transducer for the target language, but the learner has no language-specific knowledge (i.e., the learner has not access to this transducer).
- **Learning Task.** The goal of the learner is to learn a grammar for the language that will enable to produce all and only the denoting utterances for any given situation. Although the learner's representation is referred as a grammar, it does not take the form of a classical grammar from formal languages. The learner's grammar has three main components: 1) Weighted co-occurrence graph; 2) general forms; 3) decision trees. Detailed information about the construction of the learner's grammar can be found in [3].

Regarding the experimental results, the model was tested with limited sub-languages of ten different natural languages: English, German, Greek, Hebrew, Hungarian, Mandarin, Russia, Spanish, Swedish, Turkish.

In the experiments, each situation has two objects and one binary relation between them (above/below, to the left to/to the right to). Every object has

three attributes: *form, color* and *size*. The attribute of shape has six possible values (circle, square, triangle, star, ellipse, hexagon), that of color has six possible values (red, orange, yellow, green, blue, purple), and that of size three possible values (big, medium, small). Thus there are 108 different objects and 23,328 different situations. The total number of possible meanings is 113,064.

Two different measures are used to evaluate the performance of the learner: 1) *correctness*, which is the sum of the probabilities of the learner's sentences that are in the set of sentences that denote correctly one object; and 2) *completeness*, which is the fraction of sentences that denote correctly one object and appear in the set of learner sentences. A learner achieves a level p of performance if correctness and completeness are at least p. In the experiments, teacher and learner interact until the learner achieves a level $p = 0.99$.

Table 1 shows the number of interactions that are necessary to achieve this level of performance in all the languages taken into account. Each entry is the median of 10 trials. The learner is evaluated after receiving 100 sentences from the teacher.

Table 1. Number of interactions that the learner needs to reach different levels of performance. Extracted from [3]

Level	0.60	0.70	0.80	0.90	0.95	0.99
English	200	200	300	400	500	700
German	200	300	300	400	550	800
Greek	400	500	700	1500	2200	3400
Hebrew	200	300	400	500	650	900
Hungarian	200	300	350	450	550	750
Mandarin	200	200	300	400	500	700
Russian	450	500	850	1750	2350	3700
Spanish	200	300	350	500	600	1000
Swedish	200	300	300	400	600	1000
Turkish	200	200	300	400	550	800

In these results, we can distinguish two different groups: 1) Greek and Russian that need at least 3.400 interactions with the teacher; 2) the rest of languages that need at most 1.000 interactions.

4.2 Measuring Linguistic Complexity with Grammatical Inference

The model developed by Angluin and Becerra [3] proposes a unique algorithm to learn any of the languages analyzed. The system calculates the number of interactions that are necessary to achieve a good level of performance in the chosen language and shows that not all the languages need the same number of linguistic interactions to reach the same level of performance.

Taking into account the general features of the model as well as its motivation, we claim that the grammatical inference model proposed by Angluin and Becerra may be a potential adequate tool to measure the linguistic complexity

in *relative* terms. It is a model for relative complexity because its goal is to calculate the *cost* –in terms of the number of interactions needed to learn the language– to reach a good level of performance in a language. Being interested in cost, the model refers to *difficulty*, since a task that demands a large number of resources is experienced as difficult.

Relative complexity models of complexity face the problem of choosing one among the different tasks language involves and, of course, one of the many types of agents related to language. In our case, the motivation of the model forces to choose *first-language acquisition* as the task and *first-language acquirer* as the user.

The proposed grammatical inference model measures relative complexity of languages related to first-language acquisition in the following way:

- The algorithm used in the model could be equivalent to the innate capacity that allows humans to acquire a language. Therefore, the same algorithm is used to acquire any of the natural languages considered.
- The learner –this is, the machine– has no previously knowledge about the language. It has just the capacity –algorithm– to learn, but no linguistic structure previously storaged in order to facilitate the process. The machine represents, therefore, the child that has to acquire a language by just being exposed to this language.
- The model counts the needed number of interactions for the machine to achieve a good level of performance in a specific domain of the language. This is equivalent to calculate the child's cost/difficulty to acquire a language.
- The model shows that with the same algorithm not every language requires the same number of interactions. This is equivalent (in terms of complexity) to show that, with the same innate capacity, the difficulty/cost to acquire different languages is not the same and, therefore, languages differ in relative complexity.

The model, as it is designed, could give an account of *local* (no global) complexity of languages. Up to now, experiments have concentrated in a small subsystem of the language and, therefore, the model can just calculate local complexity. However, this limitation, makes possible to avoid the two general problems that any metric of global complexity must deal with: *representativity* and *comparability* [19].

Moreover, the design of the model allows to calculate complexity in *structural* terms (no system complexity). Since, the miniature language acquisition task focuses in a reduced number of structures, the model, at this moment, can just evaluate the complexity of concrete realizations.

In terms of Pallotti's classification [22], our model refers to *developmental complexity*, since it reflects the way linguistic structures emerge and are mastered in first language acquisition.

One of the main advantage of the proposed model is to offer experimental results based on exactly the same type of structures for every language. The empirical testing of language complexity and the possibility of using the same

type of data for all the analyzed languages is very important in this research field, as stated by Juola [13]: "a first requirement for any empirical test of these theories is a suitable testbed; a collection of the same messages (as far as possible) expressed in different languages."

5 Conclusion

We have presented a very preliminary approach to the study of linguistic complexity with tools coming from the field of Grammatical Inference. Our proposal is an interdisciplinary solution that uses a computational model to quantify the cost/difficulty in the process of the acquisition of different languages, showing that it is not identical in all of them.

This model is an alternative to the methods that have been used so far to calculate linguistic complexity and presents, with respect to them, the following advantages: its *interdisciplinarity*, the model combines ideas from linguistics with computational models; its *motivation*, the tool we propose is a computational model based on how humans acquire language; its *results*, it provides quantifiable experimental results; its ability to perform *crosslinguistic analysis*.

As Sinnemäki [24] emphasizes, "complexity is and has been a controversial concept in Linguistics, being simultaneously friend and foe". In fact, until recently, complexity has not been widely researched in the area of Linguistics. As pointed out by Weston [27], "though the claim that all languages are equal has been in the literature for more than 50 years, discussion of the claim has been oddly lacking. There hasn't even been any clarification of what the claim actually means. It isn't clear from the literature how the ?equality? of languages ought to be measured, let alone any investigation of whether it is true or false". Given this situation, Linguistics has to face again the problem of the complexity of natural languages. Linguists must propose tools for the analysis of linguistic complexity.

"Complexity is a central notion in linguistic theorizing and description" [24] and, therefore, results obtained from studies on linguistic complexity may have important implications both from a theoretical and from a practical point of view. From a theoretical perspective, the goal is to prove or reject one of the basic axioms in Linguistics. From a practical point of view, the analysis of complexity levels can be helpful in fields such as the teaching/learning of second languages or natural language processing.

In the future, we expect to further develop our model in order to calculate differences in languages in terms of *global* and *system* complexity. Given the referred characteristics of the proposed model, we think that to go deeper in its implementation and analysis could help us to show that languages vary in their level of complexity; that it is possible to design tools to quantify the linguistic complexity; and that the difference in the level of complexity of languages -and its measurement- is relevant to the understanding of natural languages.

References

1. Andrason, A.: Language Complexity: An Insight from Complex-System Theory. International Journal of Language and Linguistics **2**(2), 74–89 (2014)
2. Angluin, D.: Learning Regular Sets from Queries and Counterexamples. Information and Computation **75**, 87–106 (1987)
3. Angluin, D., Becerra-Bonache, L.: A Model of Semantics and Corrections in Language Learning. Technical Report, Yale University (2010)
4. Bane, M.: Quantifying and measuring morphological complexity. In: Chang, Ch., Haynie, H. (eds.) Proceedings of the 26th West Coast Conference on Formal Linguistics, pp. 69–76. Cascadilla Proceedings Project, Somerville (2008)
5. Blache, P.: A computational model for linguistic complexity. In: Bel-Enguix, G., Dahl, V., Jiménez-López, M.D. (eds.) Biology, Computation and Linguistics. New Interdisciplinary Paradigms, pp. 155–167. IOS Press, Amsterdam (2011)
6. Dahl, O.: The Growth and Maintenance of Linguistic Complexity. John Benjamins, Amsterdam (2004)
7. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars. Cambridge University Press, Cambridge (2010)
8. Deutscher, G.: Overall complexity: a wild goose chase? In: Sampson, G., Gil, D., Trudgill, P. (eds.) Language Complexity as an Evolving Variable, pp. 243–251. Oxford University Press, Oxford (2009)
9. Edmonds, B.: Syntactic Measures of Complexity. PhD.diss., University of Manchester (1999)
10. Gold, E.M.: Language Identification in the Limit. Information and Control **10**, 447–474 (1967)
11. Hawkins, J.: An efficiency theory of complexity and related phenomena. In: Sampson, G., Gil, D., Trudgill, P. (eds.) Language Complexity as an Evolving Variable, pp. 252–268. Oxford University Press, Oxford (2009)
12. Hockett, Ch.F.: A Course in Modern Linguistics. Macmillan, New York (1958)
13. Juola, P.: Assessing linguistic complexity. In: Miestamo, M., Sinnemki, K., Karlsson, F. (eds.) Language Complexity: Typology, Contact, Change, pp. 89–108. John Benjamins, Amsterdam (2008)
14. Kortmann, B., Szmrecsanyi, B.: Linguistic Complexity: Second Language Acquisition, Indigenization, Contact. De Gruyter, Berlin (2012)
15. Kusters, W.: Linguistic Complexity: The Influence of Social Change on Verbal Inflection. LOT, Utrecht (2003)
16. McWhorter, J.: The World's Simplest Grammars are Creole Grammars. Linguistic Typology **6**, 125–166 (2001)
17. Miestamo, M.: On the feasibility of complexity metrics. In: Krista, K., Sepper, M.M. (eds.) Finest Linguistics. Proceedings of the Annual Finish and Estonian Conference of Linguistics, pp. 11–26. Tallinna Ülikooli Kirjastus, Tallinn (2006)
18. Miestamo, M.: Grammatical complexity in a cross-linguistic perspective. In: Miestamo, M., Sinnemki, K., Karlsson, F. (eds.) Language Complexity: Typology, Contact, Change, pp. 23–42. John Benjamins, Amsterdam (2008)
19. Miestamo, M.: Implicational hierarchies and grammatical complexity. In: Sampson, G., Gil, D., Trudgill, P. (eds.) Language Complexity as an Evolving Variable, pp. 80–97. Oxford University Press, Oxford (2009)
20. Miestamo, M., Sinnemäki, K., Karlsson, F.: Language Complexity: Typology, Contact, Change. John Benjamins, Amsterdam (2008)

21. Newmeyer, F.J., Preston, L.B. (eds.): Measuring Grammatical Complexity. Oxford University Press, Oxford (2014)
22. Pallotti, G.: A Simple View of Linguistic Complexity. Second Language Research **31**, 117–134 (2015)
23. Sampson, G., Gil, D., Trudgill, P.: Language Complexity as an Evolving Variable. Oxford University Press, Oxford (2009)
24. Sinnemäki, K.: Language Universals and Linguistic Complexity. Three Case Studies in Core Argument Making. PhD.diss., University of Helsinki (2011)
25. Stolcke, A., Feldman, J.A., Lakoff, G., Weber, S.: Miniature Language Acquisition: A Touchstone for Cognitive Science. Cognitive Science, 686–693 (1994)
26. Valiant, L.G.: A Theory of the Learnable. Communication ACM **27**, 1134–1142 (1984)
27. Weston-Taylor, L.: Are all languages equal? PhD.diss., University of Otago (2013)

A Proposal for Contextual Grammatical Inference

Leonor Becerra-Bonache(✉), María Galván, and François Jacquenet

Laboratoire Hubert Curien, Jean Monnet University, 18 rue Benoit Lauras, 42000 Saint-Etienne, France
{leonor.becerra,maria.galvan,francois.jacquenet}@univ-st-etienne.fr

Abstract. Grammatical Inference deals with the learning of formal languages from data. Research in this field has mainly reduced the problem of language learning to syntax learning. Taking into account that the theoretical results obtained in Grammatical Inference show that learning formal languages only from syntax is generally hard, in this paper we propose to also take into account contextual information during the language learning process. First, we review works in the area of Artificial Intelligence that use the concept of context, and then, we present the theoretical, algorithmic and practical aspects of our proposal.

Keywords: Grammatical inference · Context · Natural language

1 Introduction

Grammatical Inference (GI) is a specialized subfield of Machine Learning that deals with the learning of formal languages from a set of data. Roughly speaking, in a GI problem, we have a learner (or learning algorithm) that learns a language from the information that a teacher provides to it. Hence, a GI problem has some similarities with the problem of children's language acquisition: a child, like the learning algorithm, learns a language from the data that he/she receives from the linguistic community that is around him/her. Moreover, depending on the input that the child receives, he/she will learn one language or another, that is why a child growing up in a linguistic community that speaks Spanish acquires Spanish, but if the language spoken by the community is Chinese, the child will learn Chinese.

The initial theoretical foundations of GI were given by E.M. Gold at the end of 60's [20]. His goal was to formalize the acquisition of natural language. Since then, a big amount of research has been done to establish a theory of GI, to find efficient methods for inferring grammars, and to apply those methods to practical problems (e.g., syntactic pattern recognition, adaptive intelligent agents, computational biology, natural language processing, etc.). An excellent survey of the field can be found in [15].

Most of the research that has been developed within the field of GI has focused on *syntax learning* (i.e., on learning the rules of a grammar), and tends

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 18–28, 2015.
DOI: 10.1007/978-3-319-19258-1_2

to omit any other kind of information. The theoretical results obtained in GI show that learning formal languages only from syntax is generally hard [15,20], usually leading to negative results. What about taking into account another kind of information during the language learning process?

Linguistic and cognitive studies suggest that *contextual information* seems to play an important role in the early stages of children's linguistic development [14]. For instance, let us take a conversation extracted from the CHILDES database [24] (i.e., Child Language Data Exchange System, which collects transcripts of child-parent dialogs):

Abe: milk. milk.
Father: you want milk?
Abe: uh-huh.
Father: ok. Just a second and I'll get you some.

The child in this conversation is two and a half years old, and she is concretely in the linguistic stage called two-word stage, in which children go from the production of one word to the combination of two elements. Specially in this stage, the correspondence between sentences and the context in which they are made seems to be a very important source of information for both: the child, trying to learn the language, and the adult, trying to make sense of the imperfect sentences produced by the child. For example, in the previous conversation, the child produces only two words as *milk milk* to express a whole sentence like *I want milk*. Thanks to the *context* in which the sentence is produced (which is shared by both the adult and the child), the adult can understand the meaning of the child's sentence, although it is not syntactically correct.

Therefore, contextual information seems to play an important role in language acquisition. The presence of this kind of information not only seems to facilitate (i.e. to speed-up) the learning process to the child, but also allows the communication between adult and child [14].

Taking into account that, in natural situations, contextual information is also available to the child, and that GI studies show that learning from only syntax is hard, the following questions arise: Why not to take contextual information into account during the learning process to improve language learning? Can contextual information simplify (speed-up) the learning problem?

The remainder of the paper is organized as follows. An overview of different works developed in the field of Artificial Intelligence that take into account the context is presented in Section 2. This section is divided into two subsections: Section 2.1 is dedicated to exploring different existing theories about context, and Section 2.2 reviews works on context modeling. Then, Section 3 presents the theoretical, algorithmic and practical aspects of our proposal. Finally, concluding remarks are presented in Section 4.

2 Context in Artificial Intelligence

2.1 Theories of Context

The notion of formalized contexts was introduced by McCarthy [26] in his Turing Award Lecture in the late 80's as a way of focussing the problem of generality in Artificial Intelligence. Thenceforth, it has been extensively discussed in Artificial Intelligence and other fields. Three main formalizations have been developed: *Propositional Logic of Context* (PLC), *Local Models Semantics/MultiContext Systems* (LMS/MCS) and *Situation Theory*. Next we briefly review these approaches and related papers.

Propositional Logic of Context. McCarthy [27] worked to formalize context and to develop a theory of introducing context as formal objects. He introduced a new modality $ist(c, p)$ (pronounced as "is true"), meaning that the proposition p is true in the context c. Guha's [23] PhD dissertation under McCarthy's supervision was the first in doing a depth study of context. Guha extended McCarthy's notion of context and motivated the CYC[1] ontology together with D. Lenat [22]. Knowledge statements in CYC were divided into microtheories, which become a common sense knowledge base. Each microtheory contains a set of axioms or rules and a vocabulary which provides the syntax and semantics with predicates and functions.

From these works, Buvač and others developed a *Propositional Logic of Context* (PLC) [10,11] and a *Quantificational Logic of Context* [9,25]. They described the syntax and semantics of a general propositional language of context. For this, they introduced $ist(c, p)$, meaning that the sentence p holds in the context c, and each context has its own vocabulary, i.e. a set of propositional atoms which are defined or meaningful in that context.

Local Models Semantics/MultiContext Systems. Working under a different approach, Giunchiglia [18] reformulated the problem of context in terms of locality of reasoning. He devised a formalization of context based on the problem of locality. It consists in a problem of modeling reasoning using a subset of information that reasoners know about world. Namely, to solve a problem set in a concrete situation, people do not use all their knowledge; by contrast, it is solved by a *local theory* considering who really know all the essential events of the problem. This proposal differs from reasoning, since people can move between contexts, changing from one to another when necessary. Definitively, this approach gave more importance to *formalize contextual reasoning* than *formalizing contexts as first-class objects*.

In *MultiContext Systems* (MCS), devised by Giunchiglia and Serafini [19], they introduced the idea of bridging rules. These rules relate knowledge among other contexts. MCS later became associated with the *Local Model Semantics* proposed by Ghidini and Giunchiglia [17]. Ghidini and Giunchiglia argued that

[1] CYC: http://www.cyc.com/

there are two main principles underlying the use of context: the *principle of locality* (reasoning uses a part of the environment that is available, which is called context) and the *principle of compatibility* (different kinds of reasoning performed in an environment are compatible).

Situation Theory. *Situation Theory* is a theory of meaning and communication in some situations which are recognized as primary events (as opposed to derivatives). Barwise and Perry [6] argued that for an expression to be meaningful, it should transmit information. They assumed that people use language in limited parts of the world to discuss other limited parts of the world (situations). So, they developed a theory of situations as a relation between these situations. The theory provides a system of abstract objects that makes it possible to describe the meaning of both expressions and mental states in terms of the information they carry about the external world [16].

The main ideas of *Situation Theory* are infons and situations. Infons are the basic units of information and they are of the form: $<< P, a_1, ..., a_n, i >>$ where P is an n-place relation, the value i represents polarity and $a_1, ..., a_n$ are appropriate objects for P.

Following Barwise [7], Akman and Surav treated context as "an amalgamation of a grounding situation and the rules which govern the relations within the context" [5,31].

2.2 Examples of Context Modeling

Several authors have tried to model the context. Here we present an overview of some of these works.

Nickles and Rettinger [28] introduced a *Relational Reinforcement Learning* (RRL) approach. It uses symbolic interaction between artificial agents and humans for learning denotational concept semantics. This framework does not impose any specific kind of formal context. Results were mostly positive and showed the applicability and significance of the learning framework for realistic semantic learning tasks. They could show that the agent: (a) can learn different meanings of a concept, (b) scales to different levels of complexity, including a very ambitious 10 blocks world, (c) uses communication to ease the task, (d) deals with non-stationarity and more complex scenarios.

Goldwasser and Roth [21] were interested in providing a way for a human teacher to interact with an automated learner using natural instructions, thus allowing the teacher to communicate the relevant domain expertise to the learner without necessarily knowing anything about the internal representations used in the learning process. They evaluated their approach by applying it to the rules of the solitaire card game. They showed that their learning approach can eventually use natural language instructions to learn the target concept and play the game legally. Furthermore, they showed that the learned semantic interpreter also generalizes to previously unseen instructions. This presents an analogy to human learning, where a learner tests her understanding in an actionable setting. Such a setting can be simulated as a world environment in which the linguistic

representation can be directly executed by a computer system. For example, in semantic parsing, the learning goal is to produce and successfully execute a meaning representation. Executable system actions include access to databases of simulated card games.

Chen [13] focused his research on how to build systems that use both text and the perceptual context in which it is used in order to learn a language. He first presented a system that can describe events in RoboCup 2D simulation games by learning only from sample language commentaries paired with traces of simulated activities without any language-specific prior knowledge. By applying an EM-like algorithm (Expectation-Maximization algorithm), the system was able to simultaneously learn a grounded language model as well as align the ambiguous training data. The goal of this research was to learn to map words and phrases into logical components that can be composed together to form complete meanings. Chen studied the problem in simulated environments that retained many of the important properties of a dynamic world with multiple agents and actions while avoiding many of the complexities of robotics and computer vision. His commentator system learned to semantically interpret and generate language in the RoboCup soccer domain by observing an on-going commentary of the game paired with the evolving simulator state.

The process that determines the emotional tone in a series of words is known as *Sentiment Analysis*. It is used to try to understand the attitudes, opinions and emotions expressed by a writer in an online publication. Context is an instrument to one of the existing methods for carrying out this process, as it can be seen in the paper written by Vanzo et al. [33]. They analysed the feelings in the social network (concretely Twitter). They considered a tweet within its context, i.e. the stream of related posts. In this case they studied three types of contextual information for a target tweet: an explicit conversation, the user attitude and the overall set of recent tweets about a topic. They considered the sentiment prediction as a sequential classification task over a context that associates tweets and for this, they proposed the SVM^{hmm} algorithm. Finally, they concluded that contextual information is relevant because it eliminates the ambiguity of the sentiment polarity.

3 Our Proposal

As we have pointed out previously, research in the field of GI does not take into account some relevant aspects of natural language learning like, for example, the availability of contextual information to the child. In this paper we propose to exploit syntactic and contextual information for language learning. In order to do that, we are going to focus on three different aspects: theoretical, algorithmic and practical aspects.

3.1 Theoretical Aspects

Three important formal models of language learning have been widely investigated in the field of GI.

- *Identification in the limit*, proposed by E.M. Gold [20]. In this model, examples of an unknown language are presented to the learner (or learning algorithm), and the learner has to produce a hypothesis of this language. Its hypothesis is updated after receiving each example; if the new examples received are not consistent with the current hypothesis, it changes its hypothesis. According to Gold's model, the learner identifies the target language *in the limit* if after a finite number of examples, the learner makes a correct hypothesis and does not change it from there on. There are two traditional settings within this model: a) learning from text, where only examples of the target language are given to the learner (i.e., only positive data); b) learning from informant, where examples that belong and do not belong to the target language are provided to the learner (i.e., positive and negative information).
- *Query learning model*, proposed by D. Angluin [1]. In this model, also known as active learning model, the learner is allowed to *interact* with the teacher, by making questions about the strings of the language. There are different kinds of queries, but the standard combination to be used are: a) *membership queries*: the learner asks if a concrete string belongs to the target language and the teacher answers "yes" or "no"; b) *equivalence queries*: the learner asks if its hypothesis is correct and the teacher answers "yes" if it is correct or otherwise gives a counterexample. According to Angluin's model, the learner has successfully learnt the target language if it returns the correct hypothesis after asking a finite number of queries.
- PAC learning model, proposed by L.G. Valiant [32]. The Probably Approximately Correct (PAC) model is a probabilistic model of learning from random examples. According to this model, the learner receives examples sampled under an unknown distribution. The learner is required to learn under any distribution, but exact learning is not required (since one may be unlucky during the sampling process). A successful learning algorithm is one that with high probability finds a grammar whose error is small.

As we can see, each of these models is based on different learning settings (what kind of data is used in the learning process and how these data are provided to the learner) and different criteria for a successful inference (under what conditions we say that a learner has been successful in the language learning task). Although they have aspects that make them useful to study the problem of natural language acquisition to a certain extent, other aspects make them unsuitable for this task. For example:

- in Gold's model, the definition of identification in the limit postulates greatly idealized conditions, as compared to the real-world conditions under which children learn language. Moreover, it makes some assumptions that are controversial from a linguistic point of view. For instance, there is not limit on how long it can take the learner to guess the correct language (but children are able to learn language in an efficient way), the learner hypothesizes complete grammars instantaneously (this is not the case in children's language

acquisition), and the learner passively receives strings of the language (but children also interact with their environment).
- in Angluin's model, the queries are quite unnatural for real learning environments (a child will never ask if his/her grammar is the correct one). Moreover, the learner has to learn exactly the target language (but everybody has imperfections in their linguistic competence) and the teacher is assumed to be perfect (i.e., he knows everything and always gives the correct answers), which is an ideal teacher that does not occur in a real situation.
- in Valiant's model, the requirement that the examples have the same distribution throughout the process is too strong for practical situations.

Moreover, a common feature of all these formal models is that they do not take into account some relevant aspects of natural language learning, like for example, the availability of contextual information to the child. These models take into account only the syntax; so, their goal is to learn the rules of a grammar by taking just into account if, for example, a string belongs or not to the language, but not the context in which this string has been produced. Therefore, we claim that the existing models in GI are not well adapted to introduce contextual information.

Due to the absence of formal models in GI that consider this natural aspect during the learning process, one of our goals is to develop a new theoretical framework that takes syntactic and contextual information into account for language learning.

To reach this goal, we will need to answer some crucial questions. First, how should we represent contextual information? As we have seen in the previous section, there exists different possibilities. We will opt for representing contextual information in the form of a logical language. We will explore various fields of logical languages ranging from some old formalisms such as (constraint) first order logic to more recent ones such as description logic, widely used nowadays in the area of semantic web, leading to the design and use of ontologies.

Second, how should we prove that taking contextual information into account during the learning process can not only make learning easier, but also faster? We postulate that the use of contextual information will reduce the number of data necessary to theoretically learn the target language. In other words, we think that we should be able to drive tighter generalization bounds to this additional and meaningful information. Moreover, these theoretical results should also allow us to well understand why we obtained better results with contextual information, and answer the following questions: How contextual information speeds up the learning process? What guarantees the convergence of the algorithm?, etc.

This theoretical study will thus give us a framework to develop learning algorithms that use syntactic and contextual information during the learning process.

3.2 Algorithmic Aspects

Ideally, in order to correctly simulate some aspects of natural language acquisition, the information provided to the algorithm should be the same as the one available to children. As we have seen, contextual information not only is one component of language learning, but also seems to play an important role in the first stages of children's language acquisition. Therefore, it is also of great interest to study this component and the way new learning algorithms could take it into account during the learning process.

Unfortunately, most learning algorithms in GI have been focused on syntax learning, omitting any consideration to semantics. Thus, the problem of language learning has been mainly reduced to syntax learning. For example, many algorithms in GI are based on state merging mechanisms applied on (probabilistic) automata. More precisely, these methods allow us to generalize the knowledge by merging step by step some states (which describe prefixes) while not challenging some statistical properties (e.g. in ALERGIA [12]) or the correct classification of the training data (e.g. in RPNI [29]). Therefore, this kind of algorithms are definitely not adapted to our context which has also to consider contextual information.

Our algorithmic objective is to develop new algorithms, theoretically founded, that incorporate contextual information for language learning, that is something novel in the field of GI. Our conjecture is that contextual information speeds-up the learning process and reduces the amount of examples needed to learn a language.

3.3 Practical Aspects

One of the limitations for researchers working on context is the lack of training data with contextual annotation. A common resource of training data remains conspicuously absent. Moreover, the lack of standardization of the type of data available to researchers makes it very difficult to compare different models with one another.

Our aim is to develop a new benchmark for the international community. More concretely, we will develop datasets combining natural language descriptions with semantically annotated visual events. This resource will be valuable for researchers who study language learning, particular for those who study syntax and semantics together. The datasets will be freely available for use, and the researchers will not only be able to use these data, but also produce new data and share them. Hence, these datasets aim to be collaborative tools, allowing other researchers to add new data.

This database will also allow us to evaluate our algorithms and the ones proposed by other researchers from the international community.

We can distinguish two types of useful datasets:

1) A first goal is to develop a benchmark based on the *Miniature Language Acquisition task* [30]. This task consists of learning a subset of a natural

language from sentence-picture pairs that involve geometric shapes with different properties (color, size and position). Although this task is not as complex as those faced by children, it involves enough complexity to be compared to many real-world tasks.

2) A second goal is to develop a benchmark based on real data. A first perspective is to use a software platform to develop a real syntactic-semantic dataset (in collaboration with linguists). An idea would be to present several pictures to a child, to say a sentence and ask to the child which one better reflects the content of the picture. With that, we would validate our models (we could verify, for example, if our system is able to understand what we are talking about, like the child does).

It is worth noting that our proposal relies on some preliminary works that we have done. In [2–4], it was proposed a simple computational model of language learning. This model takes into account some aspects of natural language acquisition, such as: the interaction between the adult and the child, the context in which the sentences are produced and the meaning-preserving corrections made by the adults to well understand the imperfect sentences of the child. Thus, this model has a learner and a teacher that interact in a sequence of situations by producing sentences that intend to denote one object in each situation. The learner uses cross-situational correspondences to learn to comprehend and produce denoting sentences in a given situation (there is no explicit semantic annotation of the sentences). The goal of the learner is to be able to produce every sentence denoting one object in any given situation. It was empirically showed that the access to semantic information facilitates language learning, and the presence of corrections by the teacher has an effect on language learning by the learner. Following this line of research, a work based on pair-Hidden Markov Models was proposed in [8]. It was showed that, by taking into account semantics, it is possible to accelerate the language learning process. Therefore, these preliminary works constitute the building-block of our proposal.

4 Conclusion

One of the goals of Artificial Intelligence is to enable computers to interact with the real world. To achieve this goal, we need to construct machines that are able to understand, among other things, natural language sentences addressed to them and also to produce meaningful sentences in a given context. This paper presents *ongoing work* in this line of research.

The field of GI provides a good theoretical framework for investigating the process of natural language learning. Taking into account that studies in GI are based exclusively on learning syntax, we have proposed in this paper to exploit syntactic and contextual information for language learning. Hence, our proposal establishes a methodological break up with conventional approaches in GI.

Our work can contribute to the definition and implementation of models that may simplify and improve human-computer interfaces. The results we expect to

obtain from our work are not only theoretical or restricted to the field of GI, they are also related to cognitive science, and in general, to the topic of how humans actually process natural language. So, our proposal is an interdisciplinary work and the expected results may be useful in different research areas.

References

1. Angluin, D.: Learning Regular Sets from Queries and Counterexamples. Information and Computation **75**, 87–106 (1987)
2. Angluin, D., Becerra-Bonache, L.: Learning meaning before syntax. In: Clark, A., Coste, F., Miclet, L. (eds.) ICGI 2008. LNCS (LNAI), vol. 5278, pp. 1–14. Springer, Heidelberg (2008)
3. Angluin, D., Becerra-Bonache, L.: A model of Semantics and Corrections in Language Learning. Technical Report YALE/DCS/TR-1425, Computer Science Department, Yale University (2010)
4. Angluin, D., Becerra-Bonache, L.: Effects of meaning-preserving corrections on language learning. In: International Conference on Computational Natural Language Learning (CONLL), pp. 97–105 (2011)
5. Akman, V., Surav, M.: The use of Situation Theory in Context Modeling. Computational Intelligence **13**(3), 427–438 (1997)
6. Barwise, J., Perry, J.: Situations and Attitudes. MIT Press, Cambridge (1983)
7. Barwise, J.: Conditionals and conditional information. In: Traugott, E.C., Ferguson, C.A., Reilly, J.S. (eds.) On Conditionals, pp. 21–54. Cambridge University Press, Cambridge (1986)
8. Becerra-Bonache, L., Fromont, E., Habrard, A., Perrot, M., Sebban, M.: Speeding up Syntactic Learning Using Contextual Information. International Colloquium on Grammatical Inference (ICGI) **21**, 49–53 (2012)
9. Buvač, S.: Quantificational logic of context. In: National Conference on Artificial Intelligence (AAAI), Portland, Oregon, 4–8 August 1996, vol. 1, pp. 600–606 (1996)
10. Buvač, S., Mason, I.: Propositional logic of context. In: National Conference on Artificial Intelligence (AAAI), pp. 412–419 (1993)
11. Buvač, S., Buvač, V., Mason, I.: Metamathematics of Contexts. Fundamenta Informaticae **23**(3), 263–301 (1995)
12. Carrasco, R.C., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: International Colloquium on Grammatical Inference (ICGI), pp. 139–152 (1994)
13. Chen, D.L.: Learning Language from Perceptual Context. Department of Computer Sciences, University of Texas at Austin, PhD. proposal (2009)
14. Chouinard, M.M., Clark, E.V.: Adult reformulations of child errors as negative evidence. Journal of Child Language **30**, 637–669 (2003)
15. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars. Cambridge University Press, Cambridge (2010)
16. Devlin, K.: Logic and information. Cambridge University Press, New York (1991)
17. Ghidini, C., Giunchiglia, F.: Local models semantics, or contextual reasoning = locality + compatibility. Artificial Intelligence **127**(2), 221–259 (2001)
18. Giunchiglia, F.: Contextual reasoning. Epistemologia, XVI, pp. 345–364 (1993)
19. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. Artificial Intelligence **65**(1), 29–70 (1994)

20. Gold, E.M.: Language identification in the limit. Information and Control **10**, 447–474 (1967)
21. Goldwasser, D., Roth, D.: Learning from natural instructions. Machine Learning **94**(2), 205–232 (2014)
22. Guha, R.V., Lenat, D.B.: Cyc: A midterm report. AI Magazine **11**(3), 32–59 (1990)
23. Guha, R.V.: Contexts: A formalization and some applications. Stanford PhD Thesis (1991)
24. MacWhinney, B.: The CHILDES Project: Tools for analyzing talk, 3rd edn. Lawrence Erlbaum Associates, Mahwah (2000)
25. Makarios, S.: A Model Theory for a Quantified Generalized Logic of Contexts. Technical Report KSL-06-08, Knowledge Systems, AI Laboratory (2006)
26. McCarthy, J.: Generality in Artificial. Communication of the ACM **30**(12), 1029–1035 (1987)
27. McCarthy, J.: Notes on Formalizing Context. In: Proceedings of the Thirteenth International Joint Conference in Artificial Intelligence (IJCAI-1993), Chambery, France, pp. 555–560 (1993)
28. Nickles, M., Rettinger, A.: Interactive Relational Reinforcement Learning of Concept Semantics. Machine Learning **94**(2), 169–204 (2014)
29. Oncina, J., García, P.: Identifying regular languages in polynomial time. In: Bunke, H. (ed.) Advances in Structural and Syntactic Pattern Recognition, pp. 99–108. World Scientific Publishing, Singapore (1992)
30. Stolcke, A., Feldman, J.A., Lakoff, G., Weber, S.: Miniature Language Acquisition: A Touchstone for Cognitive Science. Cognitive Science, pp. 686–693 (1994)
31. Surav, M., Akman, V.: Modeling Context with Situations. In: IJCAI-95 Workshop on Modeling Context in Knowledge Representation and Reasoning, Research Report 95/11, LAFORIA, pp. 145–156 (1995)
32. Valiant, L.G.: A Theory of the Learnable. Communication of the ACM **27**(11), 1134–1142 (1984)
33. Vanzo, A., Croce, D., Basili, R.: A context-based model for sentiment analysis in Twitter. In: International Conference on Computational Linguistics (COLING), pp. 2345–2354 (2014)

How to Search Optimal Solutions in Big Spaces with Networks of Bio-Inspired Processors

José Ramón Sánchez Couso[1(✉)], Sandra Gómez Canaval[1], and David Batard Lorenzo[2]

[1] Department of Computer Systems, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain
{jcouso,sgomez}@etsisi.upm.es

[2] Department of Research Management, University of Informatics Sciences, Crta. a San Antonio de los Baños, Km. 2.5, La Habana, Cuba
dbatard@uci.cu

Abstract. Searching for new efficient and exact heuristic optimization algorithms in big search spaces currently remains as an open problem. The search space increases exponentially with the problem size, making impossible to find a solution through a mere blind search. Several heuristic approaches inspired by nature have been adopted as suitable algorithms to solve complex optimization problems in many different areas. Networks of Bio-inspired Processors (NBP) is a formal framework formed of highly parallel and distributed computing models inspired and abstracted by biological evolution. From a theoretical point of view, NBP has been proved broadly to be an efficient solving of NP complete problems. The aim of this paper is to explore the expressive power of NBP to solve hard optimization problems with a big search space, using massively parallel architectures. We use the basic concepts and principles of some metaheuristic approaches to propose an extension of the NBP model, which is able to solve actual problems in the optimization field from a practical point of view.

Keywords: Natural computing · Bioinspired computational models · Networks of bioinspired processors · Optimization metaheuristics · Distributed and parallel algorithms

1 Introduction

A branch of Natural Computing investigates computational techniques, models of computation and computational devices inspired by nature. In the last thirty years, a great interest has been devoted to several heuristic approaches inspired by natural behaviors. They have been adopted as suitable algorithms to solve optimization problems in many different areas like pattern recognition, image processing, filter modeling, machine learning, etc. The considerable development of metaheuristics can be explained by the significant increase in the

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 29–39, 2015.
DOI: 10.1007/978-3-319-19258-1_3

processing power of the computers and by the development of massively parallel architectures [5].

Some of the main issues when solving hard optimization problems are the high-dimensional (big) search space, how to provide a suitable solution using exact techniques and how it can be solved to optimality or to any guaranteed bound within a reasonable time limit. The search space increases exponentially with the problem size, making the solution not practical. Hence, searching for new efficient and exact heuristic optimization algorithms in big search spaces currently remains as an open problem.

Bio-inspired computational models are devices able to solve NP complete problems in an efficient manner by abstracting from a computational point of view, the way in which nature computes. Their bio-inspired nature makes these models specially suitable for the simulation of complex systems. Some efforts to bring these models towards the solution of the optimization problems have been introduced in Membrane Computing [21,22,24]. The integration of metaheuristics of search and Membrane Computing has proved both their efficiency and effectiveness to solve various real world problems [23].

Networks of Bio-inspired Processors (NBP) is one of the better known natural computing frameworks based on the Formal Language Theory. NBP forms a family of highly parallel and distributed computing models inspired and abstracted from the biological evolution. On the other hand, NBP resembles a pretty common architecture for parallel and distributed symbolic processing, related to the Connection Machine [11] as well as the Logic Flow paradigm [7]. Also it is closely related to the tissue-like P systems [16] in the membrane computing area [19]. NBP had been widely investigated from a theoretical point of view [9,10,12,13,17]. The reader interested in a survey of the main results regarding NBPs is referred to [2].

Nevertheless, although the NBP model is nature inspired, it is mainly motivated by mathematical and computer science goals. Bio-inspired aspects are only considered from a qualitative and syntactical perspective. We consider this research a very promising line because, until now, almost no efforts have actually been made to solve interesting problems in real domains by means of NBP, for instance hard optimization problems. As an interested reader could have found in the literature, the NBP model has an inherent declarative, and therefore, a qualitative nature. A first step to use the NBP model as a useful search algorithm could be to adapt it to quantitative realms.

This is not the first time that a (declarative) bio-inspired model of computation is adapted to quantitative problems. In [21,22,24] it is described how optimization heuristics could be integrated in the computational framework of Membrane Computer Systems to tackle optimization problems. Here the authors take advantage of the parallel and distributed structure of this model to define a parallel optimizer by including heuristic optimizers in the membranes. However, we are actually interested in extending NBPs with numerical capabilities, keeping all the power of the (more declarative) original NBPs; and exploring its suitability for solving practical complex optimization problems within big search

spaces. This paper is organized as follows: Section 2 briefly introduces NBPs and describes the main characteristic of nature inspired optimization heuristics. Section 3 presents a theoretical discussion of our proposal to adapt the NBP framework for the solution of optimization problems. Finally, Section 4 discusses our future plans and current conclusions.

2 Theoretical Background

This section introduces briefly the NBP model and describes the main characteristics of the optimization heuristics inspired by nature.

2.1 Networks of Bio-Inspired Processors (NBP)

A Network of Bio-inspired Processors (NBP) consists of several processors, each one of which is placed in a node of a virtual graph. Each processor node acts on local data in accordance with some predefined rules and, in fact, data in every processor becomes a mobile agent which can navigate the network following a given protocol. The data travel between nodes by means of a communication process. Note that this process may require to satisfy some other conditions that are imposed by the processors when sending or receiving data (even simultaneously), by using a variety of filtering strategies (defined by filters or polarizations). A processor node is very simple; it can be either an evolutionary, splicing or genetic processor taking into account the operations it carries out [6,9,10,14]:

- An *evolutionary processor* performs very simple operations, namely, point mutations in a DNA sequence (insertion, deletion or substitution of a pair of nucleotides). This type of processor is only specialized for just one of these evolutionary operations. More generally, it can be viewed as a cell that contains genetic information encoded in DNA sequences which may evolve by point mutations. A network containing this type of processors is denominated *Network of the Evolutionary Processors* (NEP).
- A *splicing processor* performs the operation of recombination, which is presented in form of splicing. This operation is one of the basic mechanisms by means of which, DNA sequences are recombined under the effect of enzymatic activities. A network containing this type of processors is called *Network of the Splicing Processors* (NSP).
- A *genetic processor* has two operations: (1) *Mutation* between symbols (similar to the substitution operation in the evolutionary processors) and (2) *Pure and massive crossover* (similar to the splicing operation by taking empty contexts). These processors become part of a *Network of the Genetic Processors* (NGP).

In addition, there exists another type of processor called *polarized processor* [1,3], which is a special type of evolutionary processor having a valuation mapping as filtering process. This mapping represents the polarization of the processor in terms of an electrical charge: positive, negative or neutral, which is useful for

modeling some biological properties such as the inhibition or activation of the molecules or processes. A NBP having this type of processors is denominated *Network of Polarized Evolutionary Processors* (NPEP). For a more detailed view about all of the above networks, we refer the reader to [2].

2.2 Formal Definitions

In order to present the formal definition of a NBP, we introduce some notions. An *alphabet* is a finite and non-empty set of symbols. Any sequence of symbols from an alphabet V is called word over V. The set of all words over V is denoted by V^* and the empty word is denoted by ε. For any word $x \in V^*$, $alph(x)$ will denote the minimal alphabet W such that $x \in W^*$. In this paper, the different NBP we deal with are defined as follows:

- NPEP: $\Gamma_P = (V, U, G, N, \varphi, x_I, x_O)$
- NGP: $\Gamma_S = (V, U, G, N, x_I, x_O)$

where

- V and U are the input and network alphabet respectively, $V \subseteq U$.
- $G = (X_G, E_G)$ is an undirected graph with the set of vertices X_G and the set of edges E_G. It is called the *underlying graph* of the network.
- $N : X_G \to EP_U$ is a mapping which associates each node with the corresponding processor in a given NBP. Obviously, the type of the processor depends on the NBP, as it is defined below. Also N can be viewed as a set of processors (or nodes) in the sense $N = \{N(x) : x \in X_G\}$.
- $\varphi : U^* \to \mathbb{Z}$ is the valuation function.
- $x_I, x_O \in X_G$ are respectively the *input* and the *output* nodes.

As we said before, the mapping N acts on a node x giving the processor with its type. We have then the following processors:

- $N(x) = (M_x, \alpha_x, \pi_x)$ is a polarized processor as is defined in [3], where
  - M_x is a finite set of evolutionary rules. Each processor is "specialized" in one evolutionary operation only (insertions, substitutions or deletions).
  - $\alpha : X_G \to \{*, l, r\}$; $\alpha(x)$ gives the action mode of the rules application in the node x.
  - $\pi_x \in \{-, 0, +\}$ is the polarization of the node (negatively, neutral or positively charged, respectively).
- $N(x) = (M_x, A, PI, FI, PO, FO, \delta, \rho)$ is a genetic processor as is defined in [6], where
  - M_x is a finite set of mutation rules.
  - A is a multiset of strings over V with finite support and an arbitrary large number of copies of every string.
  - $PI, FI \subseteq V^*$ and $PO, FO \subseteq V^*$ are the input and output permitting/forbidding contexts, respectively.
  - $\delta = \{1, 2\}$, where $\delta = 1$ means that mutation rules are applied and $\delta = 2$ means that crossover rules are applied.
  - $\rho \in \{s, w\}$ defines the type of the input/output filters.

For more information and details about the formal definitions and properties of the above NBP (computational power, efficiency and descriptional complexity), we refer the reader to [3,12,14].

2.3 Optimization Metaheuristics and Algorithms

A metaheuristic is an efficient algorithm designed to solve approximately a wide range of hard optimization problems without having to deeply adapt to each problem [5]. Metaheuristics can be classified in single-solution based metaheuristics and population-based metaheuristics. Two major components are well differentiated in these approaches: the exploration (diversification) and the exploitation (intensification). The main differences between the existing metaheuristics concern the particular way in which they try to balance both components [4]. Roughly speaking, basic single-solution based metaheuristics are more exploitation oriented whereas basic population-based metaheuristics are more exploration oriented [5].

Single-solution based metaheuristics refers algorithms that starting from a single initial solution moves away from it, describing trajectories over the search space of the problem. Furthermore, they incorporate techniques that enable the algorithm to escape from local minima. This class of metaheuristic algorithms mainly encompasses the tabu search, simulated annealing, threshold accepting, variable neighborhood search, iterated local search, guided local search, GRASP, and so on.

Population-based metaheuristics stand of algorithms that starting from an initial set (i.e. a population) apply the generation of a new population and replacement of current one by other set of candidate solutions rather than with a single solution. The most popular population-based methods are Evolutionary Computation (EC) and Swarm Intelligence (SI).

3 A NBP Approach to Solve Optimization Problems

The first step to adapt NBP framework to solve hard optimization problems is to broaden the qualitative perspective to the quantitative one. Although it is true that one of the NBP (NPEP) incorporates a numerical evaluation over the data that processes, this fact is not used from a quantitative perspective but rather only qualitatively (NPEPs are only interested in distinguishing the overall charge: negative, positive o neutral of their components). This may be one of the main reasons why NBP models are not widely used to solve real problems. In spite of this, our exploration suggests some ideas to overcome this drawback. In this context, we propose two preliminary steps to define a NBP quantitative extension: first, we point out some general considerations associated with the optimization of elements, and second, we propose two inspired-metaheuristic approaches to extend NBP model (single-based solution and population-based solution).

To introduce our proposals, we start with some considerations:

- Optimization problems can be viewed as an architecture of several NBP [8], working each one as a subnetwork in a global network.
- An agent is a vector $A_i = (a_i^1, a_i^2, \ldots, a_i^n)$, where a_i^k represents a symbol/variable we pretend to get an optimal value. Some problems may require a set of agents, so in this case it can be helpful to use a matrix $(A_i)_{i=1}^N$.
- The filtering strategy of a NBP must be redefined for an elementary quantitative predicate. This predicate will compare the arriving strings at a given node and will find the best/worst agents.
- Configuration and computation of a NBP should be redefined to take into account these previous modifications.

We divide our exploration proposal to extend the NBP model through both single-based solutions and population-based solutions.

3.1 Single Solution Approach

The general idea consists in mapping a specific and not duplicate solution to a concrete node processor. The first step is using a modified NPEP $\hat{\Gamma}_P$ which generates a set of N agents. A *configuration* of $\hat{\Gamma}_P$ is a mapping $C : X_G \longrightarrow 2^{V^*}$ which associates a set of words with every node of the graph. The initial configuration of $\hat{\Gamma}_P$ is defined by $C_0^{(w_x)} = \{w_x\}$ for all $x \in X_G - \{x_O\}$, where w_x represents the set of words (initially one word only) which is placed at x. Note that, in this case, the input node x_I of $\hat{\Gamma}_P$ has not a particular relevance. For each $x \in X_G$, we consider a matrix of agents $(A_i^x)_{i=1}^N$, where each row represents a single agent $A_i^x = (a_i^{x,1}, a_i^{x,2}, \ldots, a_i^{x,n})$ and $a_i^{x,k}$ represents a symbol/variable we pretend to get an optimal value. To do so, we will use a vectorial function of weights/improvements $F = (F_1, F_2, \ldots, F_N)$, each component acting over all the words belonging to a node x, that is, $A_i^x = F_i(w_x)$. The matrix $(A_i^x)_{i=1}^N$ means the same in every node of the graph.

A configuration can change either by an evolutionary step or by a communication step. When changing by an evolutionary step, each component $C(x)$ of the configuration C is changed in accordance with the set of evolutionary rules M_x associated with the node x. Formally, it says that the configuration C' is obtained in *one evolutionary step* from the configuration C, written as $C \Longrightarrow C'$, iff

$$C'(x) = M_x(C(x)) \text{ for all } x \in X_G. \tag{1}$$

When changing by a communication step, each node processor $x \in X_G$ sends one copy of its dominant word whenever passes the output filter of x to all the node processors connected to x. Also, the processor x receives all the words sent by any node processor connected with x provided that can pass its input filter. Formally, it is said that the configuration C' is obtained in *one communication step* from configuration C, written as $C \vdash C'$, iff

$$C'(x) = C(x) \bigcup_{\{x,y\} \in E_G} \{z \in C^{(w_y)}(y) \,:\, z \in \pi(F(w_y)) \wedge \varphi(z) = \alpha_x\} \tag{2}$$

for all $x \in X_G$, where $\pi(F(w_y))$ is the set of words of the node y that improve the dominant word placed at the node x. Obviously $F(w_y)$ computes the best matrix of agents coming from the set w_y, φ is de valuation function of $\hat{\Gamma}_P$ and $\alpha_x \in \{-, 0, +\}$ is the polarization of the node x.

Then, the computation of $\hat{\Gamma}_P$ on w_x is a sequence of configurations $C_0^{(w_x)}(x)$, $C_1^{(w_x)}(x)$, $C_2^{(w_x)}(x), \ldots$, where $C_0^{(w_x)}(x)$ is the initial configuration of $\hat{\Gamma}_P$ on w_x, $C_{2i}^{(w_x)}(x) \Longrightarrow C_{2i+1}^{(w_x)}(x)$ and $C_{2i+1}^{(w_x)}(x) \vdash C_{2i+2}^{(w_x)}(x)$ for all $i \geq 0$. Note that the configurations are changed by alternative steps. By the previous development, each configuration $C_i^{(w_x)}(x)$ is uniquely determined by the configuration $C_{i-1}^{(w_x)}(x)$.

A computation as above *halts* if one of the following two conditions is satisfied:

- There exists two or more identical configurations and the *output* node (x_O), is non-empty.
- No further step is possible.

The algorithm summarizing this process is shown as following:

Algorithm 1. NBP Algorithm for Single-based Solution

1: An exploratory NBP generates the set $W = \{w_1, w_2, \ldots, w_M\}$ of words
2: A modified NPEP $\hat{\Gamma}_P$ with graph X_G and valuation function φ
3: $L \leftarrow$ List of nodes of $X_G - \{x_O\}$
4: **for each** $w_i \in W$ **do**
5: Select randomly $x_j \in L$
6: $x_j \leftarrow w_i$
7: Generate matrix $(A_i^{x_j})_{i=1}^N$ of agents
8: $L \leftarrow L - \{x_j\}$
9: **end for**
Require: In parallel way: for each $x \in X_G$
10: **while** $\hat{\Gamma}_P$ does not halt **do**
11: Generate an evolutionary step as defined in Eq. (1)
12: Update w_x for all $x \in X_G$
13: Generate a communication step as defined in Eq. (2)
14: Update matrix $(A_i^x)_{i=1}^N$ if necessary
15: **end while**
16: **return** matrix $(A_i^{x_O})_{i=1}^N$ met in x_O

3.2 Population Based Solution Approach

In this approach, we will use the behavior of evolutionary processors to model a genetic algorithm population. This view is in extreme opposite to the early single-based approach. In this case, we use a modified NGP $\hat{\Gamma}_S$ and distribute the individuals (words) of a global population along the nodes of the underlying graph X_G. Then we can consider (or not, depending on the problem) the subpopulation of each node $x \in X_G$ as a new population. We suppose there always

exist at least two individuals in the population belonging to every node x and, among these individuals, two of them are marked as seeds (or parents).

Following this idea, given a global population P, the initial configuration of $\hat{\Gamma}_S$ on P is defined by $C_0^{(P_x)}(x) = \{P_x\}$ for all $x \in X_G - \{x_O\}$, where P_x is the part of the population P which is assigned to the node x. We will assume henceforth that there are two individuals $\xi_x, \eta_x \in P_x$ acting as seeds. Note that the initial node x_I of $\hat{\Gamma}_S$ is not relevant in this process.

Now, a configuration can change either by an evolutionary step or by a communication step. In the case of an evolutionary step, each component $C(x)$ of the configuration C is changed in accordance with the set of splicing or crossover rules M_x, but these rules are only applied to the individuals (words) of P_x that are available for splicing or crossover operations. We recall that in this evolutionary step possibly may change the seeds ξ_x, η_x. Formally, the configuration C' is obtained in *one splicing step* from the configuration C, written as $C \Longrightarrow C'$, iff

$$C'(x) = C(x) \cup M_x(Q_x), \quad Q_x \subseteq P_x, \tag{3}$$

where Q_x is the set of words available for splicing or crossover operations. When changing by a communication step, each node processor $x \in X_G$ sends one copy of their best individuals (i.e the seeds ξ_x, η_x) which pass the output filter of x towards all the node processors connected to x. Similarly, the node processor x receives all the words sent by any node processor connected with x provided they pass its input filter. Formally, the configuration C' is obtained in *one communication step* from configuration C, written as $C \vdash C'$, iff

$$C'(x) = C(x) \bigcup_{\substack{\{x,y\} \in E_G \\ \xi_x, \eta_x \in P_x \\ \xi_y, \eta_y \in P_y}} \{(\xi_y, \eta_y) : \tau(\xi_y, \xi_x) = \texttt{true} \ \wedge \ \tau(\eta_y, \eta_x) = \texttt{true}\} \tag{4}$$

for all $x \in X_G$, where τ is a predicate containing the optimization rules, in this case, improving the seeds ξ_x, η_x of the population P_x. If the evaluation of the predicate τ is true then a copy of ξ_y, η_y enters to the x node, otherwise no improvement is achieved and the copy of ξ_y, η_y is lost.

Then, the computation of $\hat{\Gamma}_S$ on the population P is now the sequence $C_0^{(P_x)}(x)$, $C_1^{(P_x)}(x)$, $C_2^{(P_x)}(x), \ldots$, where $C_0^{(P_x)}(x)$ is the initial configuration of $\hat{\Gamma}_S$ on P_x, $C_{2i}^{(P_x)}(x) \Longrightarrow C_{2i+1}^{(P_x)}(x)$ and $C_{2i+1}^{(P_x)}(x) \vdash C_{2i+2}^{(P_x)}(x)$, for all $i \geq 0$.

A computation as above *halts* if one of the following two conditions is satisfied:

- There exists two or more identical configurations and the *output* node (x_O), is non-empty.
- No further step is possible.

The algorithm summarizing this process is shown below in Algorithm 2.
Finally, taking into account [12,15] that

- the classical complexity class NP equals to the family of languages decided by a NBP in polynomial time.

Algorithm 2. NBP Algorithm for Population-based Solution

An exploratory NBP generates a global population $P = \{P_1, P_2, \ldots, P_n\}$
A modified NGP $\hat{\Gamma}_S$ with graph X_G
$L \leftarrow$ List of nodes of $X_G - \{x_O\}$
for each P_i, $i = 1, 2, \ldots, n$ **do**
  Select randomly $x_j \in L$
  $P_{x_j} \leftarrow P_i$
  $L \leftarrow L - \{x_j\}$
end for
Require: In parallel way: for each $x \in X_G$
Mark the seeds ξ_x, η_x
while $\hat{\Gamma}_S$ does not halt **do**
  Select randomly two individuals $u, v \in Q_x$
    Splicing step with u and v
    Crossover step as defined in Eq. (3)
  Update the seeds ξ_x, η_x if necessary
  Generate a communication step as defined in Eq. (4)
end while
return Best population met in x_O

- a language is in P-class if and only if it is decided by a NBP in polynomial time and space.
- PSPACE equals to the family of languages decided by a NBP in polynomial length.

We can affirm, in an informal way, that the exploratory phase of our two approaches (see step 1 in the Algorithms 1 and 2) is an efficient algorithm to map the exploratory phase of optimization metaheuristic within our NBP extension. Moreover, we can affirm in an informal way, that the exploitation phase both single-based approach and population-based approach mainly includes predicates over the input and output filtering strategy to evaluate the best/worst words. Then, in the context of the efficiency of this exploitation phase we have changed computable predicates by computable predicates checked over finite sets of elements. Therefore, in an informally way, if NBP is able to solve NP complete problems in an efficient time and space, our extension also must be able to search solutions in big-space in an efficient time and an efficient use of resources.

4 Conclusions and Further Research Lines

The already established way of conceiving NBP as a suitable framework to solve real complex problems is to explore a quantitative extension of NBP model. In this paper, we have presented two preliminary steps to achieve this goal. First, we have introduced some general ideas about notions or concepts should be taken into account to define this NBP extension and two metaheuristic inspired approaches to extend NBP have been proposed, namely: a single based solution and population based solution. In these approaches, we have defined the

new components that should be considered in the NBP extension and we have introduced the formal definitions of these components. The main components added are computable predicates to check the evaluation of the agents which are computable predicates checked over finite sets of elements. Together with these components, we have proposed algorithms to represent the exploratory and exploitations phases in our NBP extension. Both, components and algorithms suggest that due NBP has amply demonstrated to be efficient in time and resources, our proposal of extension should not change this. Therefore, our exploratory and exploitation phases proposed should be able to address search solutions in big spaces. In this sense, we have explored in this paper, the expressive power of NBP to solve hard optimization problems with a big search space, using massively parallel architectures.

Our immediate further work will consist in to formalize the NBP extension, and formally to demonstrate its computational power. It is very important to compare the future results produced by this model with other similar models including metaheuristic approaches. This work will allow us to prove that NBP extension is an appropriate tool to solve actual problems of optimization from a practical point of view, for instance using computational simulators.

Acknowledgments. The last two authors acknowledge the partially supported results by Gain Dynamics, project iQuest which funding is applied to the *Fondo Europeo de Desarrollo Regional para el Fomento de Sectores Tecnológicos de la Comunidad de Madrid*, within the *Estrategia Regional de Investigación e Innovación para una Especialización Inteligente (RIS3)* in the *Programa Operativo de la Comunidad de Madrid 2014-2020*.

References

1. Alarcón, P., Arroyo, F., Mitrana, V.: Networks of Polarized Evolutionary Processors. Information Sciences **265**, 189–197 (2014)
2. Arroyo, F., Castellanos, J., Mitrana, V., Santos, E., Sempere, J.M.: Networks of Bio-inspired Processors, pp. 25–57. TRIANGLE. URV Publications (2012)
3. Arroyo, F., Gómez Canaval, S., Mitrana, V., Popescu, Ş.: Networks of polarized evolutionary processors are computationally complete. In: Dediu, A.-H., Martín-Vide, C., Sierra-Rodríguez, J.-L., Truthe, B. (eds.) LATA 2014. LNCS, vol. 8370, pp. 101–112. Springer, Heidelberg (2014)
4. Birattari, M., Paquete, L., Stutzle, T., Varrentrapp, K.: Classification of Metaheuristics and Design of Experiments for the Analysis of Components. Technical Report AIDA-01-05, Technische Universitat Darmstadt, Darmstadt, Germany (2001)
5. Boussaid, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. Journal of Information Sciences. **237**, 82–117 (2013)
6. Campos, M., Sempere, J.M.: Accepting Networks of Genetic Processors are computationally complete. Theoretical Computer Science. **456**, 18–29 (2012)
7. Errico, L., Jesshope, C.: Towards a new architecture for symbolic processing. Artificial Intelligence and Information-Control Systems of Robots, World Sci. Publ., Singapore, 31–40 (1994)

8. Gómez Canaval, S., Sánchez, J.R., Arroyo, F.: Simulating metabolic processes using an architecture based on networks of bio-inspired processors. In: Mauri, G., Dennunzio, A., Manzoni, L., Porreca, A.E. (eds.) UCNC 2013. LNCS, vol. 7956, pp. 255–256. Springer, Heidelberg (2013)
9. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Solving NP-complete problems with networks of evolutionary processors. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2084, p. 621. Springer, Heidelberg (2001)
10. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Networks of Evolutionary Processors. Acta Informática **39**, 517–529 (2003)
11. Hillis, D.W.: The Connection Machine. MIT Press A.I. Memo No. 646 (1981)
12. Manea, F., Margenstern, M., Mitrana, V., Pérez-Jiménez, M.J.: A New Characterization of NP, P, and PSPACE with Accepting Hybrid Networks of Evolutionary Processors. Journal of Theory of Computing Systems **46**, 174–192 (2010)
13. Manea, F., Martín-Vide, C., Mitrana, V.: On the Size Complexity of Universal Accepting Hybrid Networks of Evolutionary Processors. Mathematical Structures in Computer Science **17**, 753–771 (2007)
14. Manea, F., Martín-Vide, C., Mitrana, V.: Accepting networks of splicing processors: Complexity results. Journal of Theoretical Computer Science **371**(1), 72–82 (2007)
15. Manea, F., Martín-Vide, C., Mitrana, V.: Accepting Networks of Evolutionary Word and Picture Processors: A survey, pp. 523–560. Frontiers in Mathematical Linguistics and Language Theory, World Scientific (2010)
16. Martín-Vide, C., Pazos, J., Păun, G., Rodríguez-Patón, A.: A new class of symbolic abstract neural nets: tissue p systems. In: Ibarra, O.H., Zhang, L. (eds.) COCOON 2002. LNCS, vol. 2387, p. 290. Springer, Heidelberg (2002)
17. Martín-Vide, C., Mitrana, V., Pérez-Jiménez, M., Sancho-Caparrini, F.: Hybrid networks of evolutionary processors. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723. Springer, Heidelberg (2003)
18. Margenstern, M., Mitrana, V., Jesús Pérez-Jímenez, M.: Accepting hybrid networks of evolutionary processors. In: Ferretti, C., Mauri, G., Zandron, C. (eds.) DNA 2004. LNCS, vol. 3384, pp. 235–246. Springer, Heidelberg (2005)
19. Pâun, G.: Computing with membranes. Journal of Computer and System Sciences **61**, 108–143 (1998)
20. Navarrete, C., Echeandia, M., Anguiano, E., Ortega, A., Rojas, J.: Parallel simulation of NEPs on clusters. Proc. of Int. Conf. of Web Intelligence and Intelligent Agent Technology. IEEE Computer Society **3**, 171–174 (2011)
21. Nishida, T.: Membrane algorithms: approximate algorithms for NP-complete optimization problems. In: Ciobanu, G., Păun, G., Pérez-Jiménez, M.J. (eds.) Applications of membrane computing, pp. 303–314. Springer, Heidelberg (2006)
22. Song, X., Wang, J.: An Approximate Algorithm Combining P Systems and Active Evolutionary Algorithms for Traveling Salesman Problems. International Journal of Computers Communications and Control **10**(1), 89–99 (2014)
23. Zhang, G., Cheng, J., Gheorghe, M.: A membrane-inspired approximate algorithm for traveling salesman problems. Romanian Journal of Information Science and Technology **14**(1), 3–19 (2011)
24. Zhou, F., Zhang, G., Rong, H., Gheorghe, M., Cheng, J., Ipate, F., Lefticaru, R.: A particle swarm optimization based on P systems. Proc. IEEE Sixth International Conference on Natural Computation (ICNC) **6**, 3003–3007 (2010)

Distributed Simulation of NEPs Based On-Demand Cloud Elastic Computation

Sandra Gómez Canaval[1(✉)], Alfonso Ortega de la Puente[2], and Pablo Orgaz González[1]

[1] Department of Computer Systems, University College of Computer Science, Universidad Politécnica de Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain
sgomez@etsisi.upm.es, p.orgaz@alumnos.upm.es

[2] Department of Computer Systems Engineering, Universidad Autónoma de Madrid, Campus of Cantoblanco, 28049 Madrid, Spain
alfonso.ortega@uam.es

Abstract. Networks of Evolutionary Processors (NEP) are a bio-inspired computational model able to solve NP complete problems in an efficient manner. Up to now, the only way to analyze and execute these devices is through hardware and software simulators able to encapsulate the inherent parallelism and the efficiency in their computations. Nowadays, simulators for these models only cover many software applications developed under sequential/parallel architectures over multicore desktop computers or clusters of computers. Most of them, are not able to handle the size of non trivial problems within a massively parallel environment. We consider that cloud computation offers an interesting and promising option to overcome the drawbacks of these solutions. In this paper, we propose a novel parallel distributed architecture to simulate NEPs using on-demand cloud elastic computation. A flexible and extensible simulator is developed in order to demonstrate the suitability and scalability of our architecture with several variants of NEP.

Keywords: Natural computing · Cloud elastic computation · Distributed architectures · Parallel computation · Bio-inspired computational models · Networks of evolutionary processors

1 Introduction

Bio-inspired computational models are able to solve NP complete problems in an efficient manner by abstracting from a computational point of view, the way in which nature computes. Networks of Evolutionary Processors (NEP) are one

A.O. de la Puente — Results partially supported by Gain Dynamics, project iQuest which funding is applied to the *Fondo Europeo de Desarrollo Regional para el Fomento de Sectores Tecnológicos de la Comunidad de Madrid*, within the *Estrategia Regional de Investigación e Innovación para una Especialización Inteligente (RIS3)* in the *Programa Operativo de la Comunidad de Madrid 2014-2020*.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 40–54, 2015.
DOI: 10.1007/978-3-319-19258-1_4

of the most well known bio-inspired models based on the formal language theory. NEP can be defined as a graph whose nodes are processors which perform very simple operations over strings and send their results to other nodes. These operations consist in erasing a symbol, adding a symbol or replacing a symbol with other. Every node has filters that block some strings from being sent and/or received. As computing devices, NEP alternate evolutionary and communication steps, until a predefined stopping condition is fulfilled. All processors change their contents at the same time in each evolutionary step. In the communication step, the strings that pass the corresponding filters are interchanged between nodes connected together.

NEP have been widely investigated from a theoretical point of view: NEPs as language generating, accepting devices and problem solvers [4,5,12], characterization of the complexity classes NP, P, and PSPACE based on accepting NEPs [14], universal NEPs and some descriptional complexity problems [15].

Up to now, it has neither been possible to have implementations in vivo nor in vitro of many bio-inspired models including the NEP model. The only way to analyze and execute these devices is through hardware and software simulators. It is worth mentioning that several software simulators of NEPs using multithreaded Java have been reported in [6,8,9] but them only offer specific and dedicated implementations over multicore desktop computers. Additionally, a novel multithreaded Java simulator proposing the use of massively parallel platforms for multicore desktop computers and clusters of computers named jNEP was introduced in [2,16]. It is certainly true that all of these simulators are tools helping the researchers in the extraction of results from a model without the need of having a real implementation. But when the model is simulated on conventional computers, the total amount of space needed to simulate and to run their algorithm, usually becomes exponential [16]. This may be one of the main reasons which natural computers are not widely used to solve real problems. Therefore, these implementations are not necessarily suitable approaches to overcome the drawbacks of the specific hardware and software simulators running on conventional computers. However, the good results obtained by the clusters architecture proposed in [16] suggest their scalability and viability in other massively parallel platforms in order to minimize the dependencies between master and slaves clusters and therefore the bottleneck in massive communications.

In this paper we propose two contributions. First, a novel parallel distributed architecture to simulate NEPs using on-demand cloud elastic computation and second, a simulator named Nepfix based on this architecture. Nepfix is a flexible and extensible computational framework able to simulate several variants of NEP model.

This work is organized as follows. In section 2, both NEP model and a brief description about related works in distributed simulation of NEPs are introduced. Section 3 describes the distributed architecture to simulate NEPs using on-demand cloud elastic computation. Section 4 describes the design and implementation of Nepfix. Moreover, section 5 shows some performance results obtained in the simulation of a simple real example using Nepfix in order to prove

their suitability to solve real problems. Finally, in section 6, the conclusions and future work are discussed.

2 Theoretical Background and Related Work

2.1 Networks of Evolutionary Processors (NEP)

In this section, we introduce NEP model by following the definitions introduced in [4,5]. First, we start by summarizing the notions used throughout this paper. An alphabet is a finite and nonempty set of symbols. Any sequence of symbols from an alphabet V is called *word* over V. The set of all words over V is denoted by V^* and the empty word is denoted by ε. The length of a word x is denoted by $|x|$. A homomorphism from the monoid V^* into the monoid (group) of additive integers $\mathbb{Z}$ is called *valuation* of V^* in $\mathbb{Z}$.

The rules in the basic NEP model describe rewriting operations that may be viewed as linguistic formulations of punctual mutations of nucleotides. A rule $a \to b$ with $a, b \in V \cup \varepsilon$ is a *substitution rule* if both a and b are not ε; it is a *deletion rule* if $a \neq \varepsilon$ and $b = \varepsilon$ and it is an *insertion rule* if $a = \varepsilon$ and $b \neq \varepsilon$. The set of all substitution, deletion, and insertion rules over an alphabet V are denoted by Sub_V, Del_V, and Ins_V, respectively. If a rule σ is a substitution rule, it is applied in any place of the word (represented by $\alpha = *$) , if σ is a elimination or insertion rule, it can be applied ether in the left, in the right or in any place of the word ($\alpha = l$, $\alpha = r$, $\alpha = *$, respectively). α is named the *action mode* of the rules.

Definition **1** *: An evolutionary processor over V as in [5] is a 5-tuple (M, PI, FI, PO, FO), where:*

- *M is a set of substitution, deletion or insertion rules over the alphabet V. Formally, $M \subseteq Sub_V$ or $M \subseteq Del_V$ or $M \subseteq Ins_V$. M represents the set of evolutionary rules of the processor. As one can see, one processor is "specialized" in one evolutionary operation only.*
- *$PI, FI, PO, FO \subseteq V$ are the input and output permitting/forbidding contexts of the processor respectively, such that $PI \cap FI = \emptyset$ and $PO \cap FO = \emptyset$.*

As can be observed, the evolutionary processor is a mathematical concept similar to an evolutionary algorithm which are inspired from the Darwinian evolution [4]. Then, the rewriting operations are interpreted as mutations and the filtering process is viewed as a selection process.

Definition **2** *: Network of evolutionary processors (NEP) as in [4] is a 8-tuple $\Gamma = (V, U, G, N, \varphi, \beta, x_I, x_O)$ where:*

- *V and U are the input and network alphabet, respectively such that $V \subseteq U$.*
- *$G = (X_G, E_G)$ is an undirected graph without loops with the set of vertices X_G and the set of edges E_G. G is called the underlying graph of the network.*

- $N : X_G \rightarrow EP_U$ *is a mapping which associates with each node* $x \in X_G$ *the evolutionary processor* $N(x) = (M_x, PI_x, FI_x, PO_x, FO_x)$. EP_U *is the set of evolutionary processors over* U.
- $\alpha : X_G \rightarrow \{*, l, r\}$; $\alpha(x)$ *gives the action mode of the rules application in the node* x.
- $\beta : X_G \rightarrow \{s, w\}$ *defines the type of the input/output filters of a node: strong or weak, respectively. More precisely, for every node* $x \in X_G$, *the following filter are defined:*
  Input filter: $\rho(x)(\cdot) = rc_{\beta(x)}(\cdot; PI_x; FI_x)$
  Output filter: $\tau(x)(\cdot) = rc_{\beta(x)}(\cdot; PO_x; FO_x)$
- $x_I, x_O \in X_G$ *are the input and the output node of* Γ *respectively.*

In [1], a variant of NEP model named *Networks of Polarized Evolutionary Processors (NPEP)* was introduced. The main difference with regard to the NEP model is the filtering strategy based on the polarity concept. Then, a new type of processor named *evolutionary polarized processor* has a specific polarization representing an electrical charge: positive, negative or neutral. A valuation mapping defined in the network, assigns to some symbols an integer value, allowing to the processors, filter a word depending of their sign (negative, zero or positive). Recently in [3], is proved that NPEP is a computationally complete model.

Definition **3** *: A polarized evolutionary processor over* V *defined as in [3] is a 3-tuple* (M, α, π) *where:*

- M *and* α *are defined as above in the Definition 1.*
- $\pi \in \{-, 0, +\}$ *is the polarization of the node (negatively, neutral or positively charged, respectively).*

Definition **4** *: Network of polarized evolutionary processors (NPEP) as is defined in [3] is a 7-tuple* $\Gamma = (V, U, G, R, \varphi, x_I, x_O)$ *where:*

- V, U, G, x_I *and* x_O *are defined as above in the Definition 2.*
- $R : X_G \rightarrow EP_U$ *is a mapping which associates with each node* $x \in X_G$ *the polarized evolutionary processor* $R(x) = (M_x, \alpha_x, \pi_x)$.
- φ *is a valuation of* U^* *in* $\mathbb{Z}$.

The dynamic of NEP and NPEP models is given by the following notions. A *configuration* of a NEP/NPEP Γ is defined as a mapping $C : X_G \longrightarrow 2^{V^*}$. A configuration may be understood as the sets of words which are present in any node at a given moment. Given a word $w \in V^*$, the initial configuration of Γ on w is defined by $C_0^{(w)}(x_I) = \{w\}$ y $C_0^{(w)}(x) = \emptyset$ for all $x \in X_G - \{x_I\}$.

A configuration can change either by an *evolutionary step* or by a *communication step*. When changing by a an evolutionary step, each component $C(x)$ of the configuration C is changed in accordance with the set of evolutionary rules M_x associated with the node x both for NEP as for NPEP model. For NEP model, in one communication step, each node processor $x \in X_G$ sends one copy

of each word it has which can pass the output filter of x to all the node processors connected to x. And it receives all the words sent by any node processor connected with x providing that they can pass its input filter.

On the other hand, during a communication step in the NPEP model, each node processor $x \in X_G$ sends out a copy of every word with different polarity that x and receives words, coming from interconnected processor nodes, having the same polarity than x. Formally, the computation of a NEP/NPEP Γ on the input word $w \in V^*$ is the sequence of configurations $C_0^{(w)}, C_1^{(w)}, C_2^{(w)}, \ldots$, where $C_0^{(w)}$ is the initial configuration of Γ on w, $C_{2i}^{(w)} \Longrightarrow C_{2i+1}^{(w)}$ and $C_{2i+1}^{(w)} \vdash C_{2i+2}^{(w)}$, for all $i \geq 0$. Note that the configurations are changed by alternative steps (evolutionary and communications steps, respectively).

Finally, a *halting* computation both NEP and NPEP models is achieved if the following two conditions are satisfied: 1) there exists a configuration in which the set of words existing in the output node is non-empty or 2) no further step is possible. For more details about the formal definitions and properties of the NEP and NPEP models, we refer the reader to [1,3,5].

2.2 Related Works in Distributed Simulation of NEP Model

Up to now, it has not been possible to have implementations neither in vivo nor in vitro of many bio-inspired models including NEP model. The only way to analyze and execute these devices is through hardware and software simulators. Preliminary Multithreaded Java simulators for NEP model developed as specific purpose software applications have been reported in [6,8,9]. The first general purpose simulator of NEP model was introduced in [2,16] with the name jNep. jNep is a multithreaded Java simulator using massively parallel platforms for multicore desktop computers through Java RMI protocol. In this approach, the simulation is realized over the same hardware platform.

Afterwards, a novel parallel architecture based on clusters defining a master-slave structure of computers was proposed in [16,17]. In this approach, an interface based on MPI message passing, developed in C++ is used as mechanism for the communication. This simulator parallelizes any system that could be expressed as a set of logical nodes that communicate with others. The simulation is performed in a synchronous way, that is, all the nodes compute the next step of the simulation at the same time, then they share their results and are ready for the next step. The load of each computer could be balanced between two simulation steps and it is executed by a master processor in the cluster. Other tasks in the master node are: to collect results of all the processors to send to each one data needed for the next step and either to add or remove processors to the simulation in a specific moment.

An important approach for solving the bottleneck in the communications and the drawback on the distribution of a specific problem among a set of processors can be found in [10]. This approach was developed for a simulator for P-systems [19] (natural computation model inspired by membranes). The structure proposed in this approach could be sketched as follows:

- In a distributed platform, this simulator must be responsible of just a subdomain of the complete system, otherwise, if only one computer is running the system, it has to be able to do the whole work.
- Modules by splitting the domain and by communicating all the computers with each other are needed requirements to complete the architecture. The authors have mentioned that AMQP protocol (Advanced Messaging Queuing Protocol [21]) covers these requirements. AMQP provides a set of mechanisms (e.g. exchanges, queues, bindings, topics, etc) able to tackle all typical interprocess communication needs, in a general and efficient way that includes the possible access to distributed resources.
- The use of a protocol like AMQP implies the definition of the corresponding communication protocol in the simulator. This protocol must state messages by sharing information between the subdomains running by each computer and messages both to enroll and by leave the simulation. Likewise must be included messages to send the subdomain to new computers.
- Structure needs computers acting as: an *initial computer* (master) which starts the simulation; *proxy's or father nodes* which have the power to split their domain to share it with a new computer that enrolls the simulation and *leaves* which are not able to split their domains anymore.
- Each computer should have a copy of the simulator with a wrapper to handle the messages, but only master and proxy nodes need a copy of the broker running with the simulator.

Finally, we mention another important contribution to the simulation of P System that follows a hardware approach. In [13], a simulator using GPU's is described and compared with other conventional CPU's simulators. Here, it is demonstrated that GPUs are well suited to simulate P systems due to the highly parallelism that they present in its architecture but the P systems based solutions must be redesigned to be adapted to the GPU programming model in order to improve the performance on the simulations.

3 Distributed Architecture Using On-Demand Cloud Elastic Computation

In order to remark the importance and novelty of the architecture proposed in this section, the architectures and implementations of bio-inspired computers introduced in the section 2.2 are analyzed. The main drawbacks in the parallel architectures described previously are:

- The inherent difficulties both technical as financial ones associated with the access to massive parallel hardware. For instance, there is no standard JVM for clusters rather than there are several experimental approaches mainly based on the RMI protocol. These approaches usually don't focus on what we need to distribute the simulation of our systems among a set of processors.
- The master-slave structure frequently followed in these approaches is based on clusters. In [16], it is checked that in this approach, the presence of the

master becomes an important bottleneck that limits the size of the problem being solved: only small instances of interesting problems could be tackled before the system crashes due to lack of resources, and, hence, it can not be used to solve general real life problems.
– The space constraints of simulators designed to be run on just a unique, even powerful, computer. Even today, a goal is try to overcome the difficulty to provide a general purpose approach that allows to the researchers generate simulations for any instance with pre-made architectures and solutions.

All reasons mentioned above, are some of the main reasons which natural computers are not widely used to solve real problems. Therefore, the implementations introduced in section 2.2 are not necessarily suitable approaches to overcome the drawbacks of the specific hardware and software simulators running on conventional computers. In this context, the main ideas that we propose in our architecture in order to minimize these drawbacks could be summarized as follows:

– It is needed a methodology allowing that new general purpose models of computation (like P-systems or NEPs, for instance) become available to the researchers, as an alternative to classic von Neummann computers. In this sense, it is necessary to develop appropriate new frameworks to describe how the problems can be solved with these new computers.
– Providing a platform in which simulation could be performed by a dynamic set of computers that could join (and leave) at any moment the simulation.
– When more than one computer are simulating a NEP, none needs to know the whole system. The only scenario in which one computer stores the complete system is when there is only one computer solving the problem. When a new computer enrolls the system, each computer only knows the part of the systems assigned to it.
– Delegating, as much as possible, some tasks (mainly those concerning communications between computers) to well known, tested and efficient state-of-the-art tools.

It is worth mentioning that in a previous work, it is checked other similar although less general approach to the distributed simulation of P-systems. In [7], is explored the possible use of the Map Reduce paradigm by means of its implementation by Hadoop. We have concluded that, although it is possible and even could be efficient, Map Reduce is a natural way to store and query big amounts of data in a distributed way. It seems, hence, artificial to store P-systems into a distributed database and to translate the semantics of the P-system into a sequence of queries. We admit that it is a clever, but rather artificial, approach. This is why we have discarded it as the basis for a general methodology to simulate any natural computer including the NEP model.

3.1 On-Demand Cloud Elastic Architecture

In order to overcome the drawbacks detected in the existing approaches as mentioned above, we propose a novel parallel distributed architecture to simulate

NEPs using on-demand cloud elastic computation. Massive cloud platforms have demonstrated to be a suitable model for the provisioning of resources to those scientific communities that are typically excluded by the access to supercomputing infrastructures or whose experiments require a variable demand of resources thus enabling a reduction of maintenance costs [11]. We consider that the access to cloud resources on demand allows scalability, avoid bottlenecks inherent to master-slave approaches and break the space limits of simulators based on local platforms, etc. Our on-demand cloud elastic architecture consist basically in two blocks, namely: *NEP simulator* and *Spring server* and their detailed structure is shown in the Figure 1.

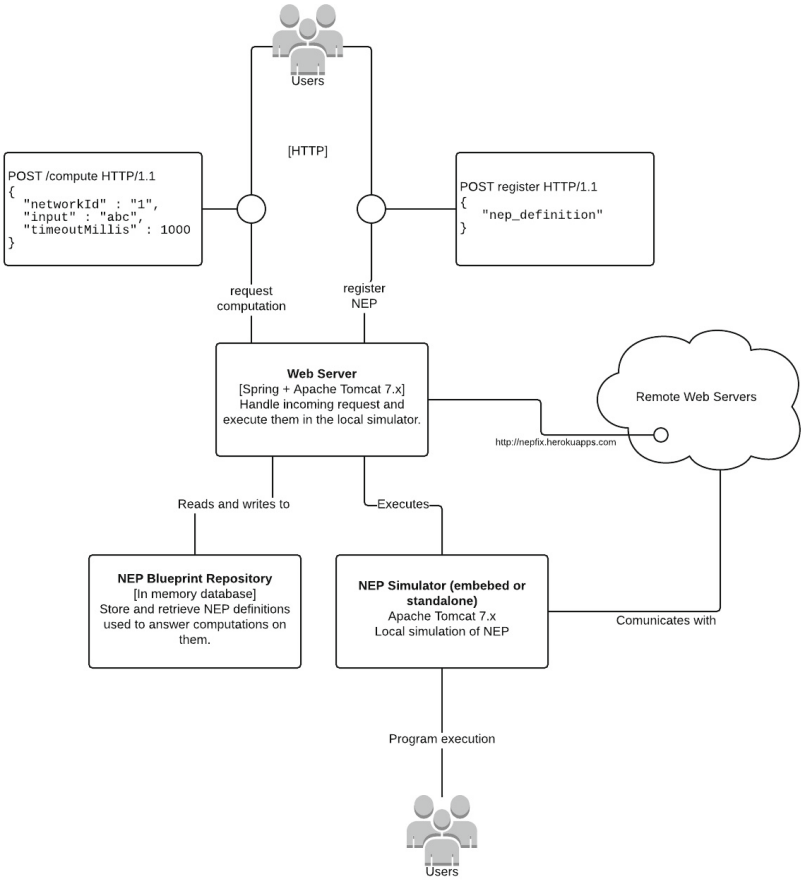


Fig. 1. Nepfix over on-demand cloud elastic architecture

NEP Simulator Block

This block contains all the code related to NEP model definition of the processes of load, execution and communication with remote NEPs via HTTP protocol.

The *Execution model process* contains the main technical aspect of our architecture: the balance between an acceptable level of performance governed by the design of the selection process responsible to emulate NEP execution and the achievement of an extensible and easily maintainable code. In order to achieve an acceptable level of performance, we have defined three novel mechanisms in this block:

- **Referential transparency in evaluations**: mechanism for controlling that processors, filters or any other NEP component that takes part in the computation, not be able to allocate any memory during the evaluation of a word and must not mutate the state of the NEP.
- **Memory overhead management based on immutability of elements**: mechanism avoiding all allocations during the execution which guarantees the immutability of the elements in the NEP. This mechanism allows bigger problems to fit into memory and essentially eliminates the need for GC (garbage-collection) pauses.
- **Definition of a local queue of computations**: mechanism controlling a queue of computations defined by future objects that are consumed by a thread pool. In NEP, every processor is executed in a parallel way. From a practical point of view, it means that each processor must be allocated in a specific core, therefore the number of cores available for the simulator must be much lower than the number of processors.

Spring Server Block
This block contains a Spring application capable of handling HTTP requests regarding in memory registration of NEPs and execution of computation requests. It depends on the simulator block and does not contain any logic related to NEPs execution apart from handling the input to the specific simulator instance. Spring server can be deployed as long as JVM is installed and a HTTP port is available. In this context, this block is able to handle any HTTP incoming request regarding computations. The flow of information to communicate with the Server block relies in plain JSON (as data format for input and output) and HTTP. Note that we do not need either database or external services to launch the server.

4 Nepfix Simulator

Nepfix is the open source project we have developed as second contribution of this work. Nepfix is a generic, extensible and scalable solution for NEPs simulation based on on-demand cloud elastic architecture proposed in the previous section. One of the main goals of Nepfix is to provide a tool to assure the feasibility of NEPs as a computation paradigm which is able to solve practical problems through traditional programs. The strategy to achieve this goal consist to delegate either specific parts or all problem to be solved to the distributed components of the platform.

Nepfix project is fully written in Java 8 and it is built using Gradle. Java allows to Nepfix support these characteristics: portability across different cloud

environments and wide support (Amazon AWS, Heroku, Pivotal, etc), the local deployment is straightforward since no installation is required apart from the Java Virtual Machine (JVM), it has an excellent community support and finally, Java has a great variety of frameworks like Spring (which is used as the HTTP facade for our simulator). Then, the blocks defined in our architecture (section 3.1), are implemented as a traditional Java program and is executed without any other dependency.

In order to provide Nepfix with a similar level of parallelism as close as possible to the NEP model, we have defined a fixed pool of threads. This pool executes instructions which are queued and processed, resulting in new instructions or an output. It is possible to execute every processor in it as an own thread using Green Threads. We remark that this implementation requires a configuration of the JVM not always available in cloud sandboxes.

The design of Nepfix follows the main quality principles of scalable and reusable software architectures. Nepfix has defined a software architecture of three layers based on Model-View-Controller pattern. In this context, interface segregation and separation are considered. The core elements of the NEP capable of performing any computation (namely, filters and processors), are abstracted by a simple interface. This interface contains only one method able to accept the condition imposed by the filtering strategy and process it, in a evolutionary way. This simple design decision allow us to maintain the Nepfix implementation concise and extensible. The UML class diagram exhibiting these software design considerations is given in the Figure 2.

Nepfix works as follows. The first step is the definition of the NEP model encoding the specific problem definition which will be executed it the same way that any traditional program. This definition is a plain file in JSON format that contains three sections: processors, filters and nodes (network). Optionally a network identifier is given since a Nepfix server may contain multiple NEPs. As we have mentioned above, in terms of Netfix code, the definition of filters and processors are equivalent. They require an id, a kind of class and optional arguments. Nepfix will instantiate both the corresponding class as the elements with the given arguments. There is no need to modify existing simulator code to add new processors or filters, only the specific element logic must be coded. Nodes of the graph are declared in a similar way. Every node must have a processor and a filter and every connection must have a filter (identity filters and processors are provided). Optionally, a node can be marked as input and a connection can be marked as output for the NEP.

To allow Nepfix to communicate with other NEPs in remote machines, nodes in the graph must be declared as remote. A remote node is conceptually the same than a local node but the processor logic is replaced by a HTTP call to remote NEP. Since this call is a separate computation, maximum results and timeout properties must be specified as well as the URL where the remote Nepfix instance is located. In order to perform the computation of this NEP, a definition path must be given to the local program or uploaded to the remote server using HTTP. Once the NEP is correctly loaded for a computation, it may be performed using

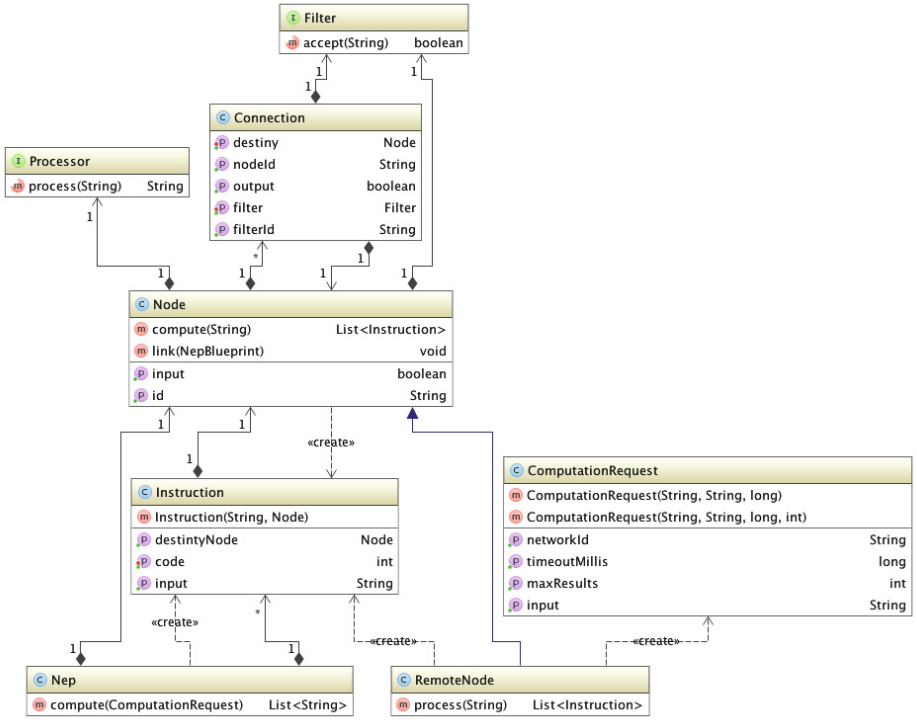


Fig. 2. Class diagram representing the most relevant classes and relationships in Nepfix

a Curl command. Finally, the complete computation is collected by Nepfix in the output node and their results are exported to the output plain text file and at the same time, are displayed in console window. Specific details about server configuration and deployment are out the scope of this paper and will not be included, however, all the relevant information lives inside the project configuration files available in [20].

5 Nepfix Solving the Fizzbuzz Problem

In order to show a basic and simple execution of Nepfix simulator in a fun real program, we propose use a modified version of FizzBuzz problem. The definition of original problem is the following: "Write a program that prints the numbers from 1 to n. For multiples of three print *fizz* instead of the number and for the multiples of five print *buzz*. For numbers which are multiples of both three and five print *fizzbuzz*" [18].

Now, we define a NPEP algorithm for the Fizzbuzz problem. Given an alphabet $\Sigma = \{a\}$ we define a NPEP Γ able to generate all $w \in \Sigma^*$ where $3 \leq |w| \leq n$ for any $n \in \mathbb{N}$ such that

$$mod(|w|, m) = 0 \begin{cases} \text{if } m = 3 & \text{printed result is the word } \textit{fizz} \\ \text{if } m = 5 & \text{printed result is the word } \textit{buzz} \\ \text{if } m = 3 \wedge m = 5 & \text{printed result is the word } \textit{fizzbuzz} \end{cases}$$

where $mod(|w|, m) = 0$, such that $m \in \mathbb{N}$, means that $|w|$ is multiple of m.

The formal definition of $\Gamma = (V, U, G, R, \varphi, x_I, x_O)$ is as follows:

- $V = \{a\}$ and $U = V \cup \{a', \bar{a}, \hat{a}, \breve{a}, \tilde{a}, b, b', c, c', f, z\}$ are the input and network alphabet respectively.
- G is the underlying graph defined as in Figure 3.
- R is a mapping as is given in Table 1.
- φ is given by:
  $\varphi(a) = 1, \varphi(b) = 3, \varphi(c) = 5, \varphi(b') = -4, \varphi(c') = -6$
  $\varphi(a') = \varphi(\bar{a}) = \varphi(\hat{a}) = \varphi(\breve{a}) = \varphi(\tilde{a}) = -1$
  $\varphi(f) = \varphi(z) = 100.$
- $x_I, x_O \in X_G$ are the input and the output node of Γ respectively.

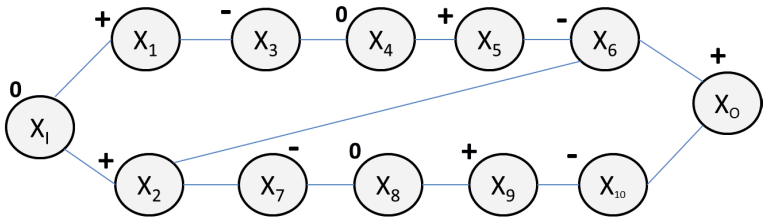


Fig. 3. Γ graph for Fizzbuzz problem

In order to generate and to evaluate the n first natural numbers, Γ must generate n number of computations, then Γ halts in the n configuration. Note

Table 1. Parameters definition of the nodes in the G

Node	M_x	α_x	**Node**	M_x	α_x
X_I	$\varepsilon \to a$	0	X_1	$a \to a', a \to \bar{a}, a \to \hat{a}$	+
X_2	$a \to a', a \to \bar{a}, a \to \hat{a}, a \to \breve{a}, a \to \tilde{a}$	+	X_3	$\varepsilon \to b$	-
X_4	$a' \to a, \bar{a} \to a, \hat{a} \to a$	0	X_5	$b \to b'$	+
X_6	$\epsilon \to f$	-	X_7	$\epsilon \to c$	-
X_8	$a' \to a, \bar{a} \to a, \hat{a} \to a, \breve{a} \to a, \tilde{a} \to a$	0	X_9	$c \to c'$	+
X_{10}	$\epsilon \to z$	-	X_O	$b' \to \varepsilon, c' \to \varepsilon$	+

that input node (x_I) generates words representing the n first natural numbers (the length of each word represent a particular $n \in \mathbb{N}$). Nodes X_1, X_3, X_4, X_5 and X_6 are the nodes generating the words which length is multiple of 3. Each one of these words is marked with the f symbol, which represents the *fizz* string. In a similar way, nodes X_2, X_7, X_8, X_9 and X_{10} generate the words which length

is multiple of 5. Note that the nodes X_2 and X_6 are connected suggesting that, this is a path followed by the words which lengths are both multiples of 3 and 5. Finally, all the words representing natural numbers between 3 and n are collected in x_O node.

In order to perform the computation of this Γ, a definition path must be given to the local program or uploaded to the remote server using HTTP. Once the Γ is correctly loaded, a computation may be performed. Using a Curl command, the computation will be requested being the number of "a" in the input the counter as is showed in the Figure 4. In this Figure, it is observed that the input word contains nine symbols a. The results obtained are: three *fizz* strings printed (corresponding to 3, 6 and 9 which are multiples of 3), one *buzz* string printed (corresponding to 5 as multiple of himself) and no *fizzbuzz* string printed. Note that the output printed the results which are obtained in an asynchronus way which is a characteristic of Nepfix. Complete implementation of Γ can be found in the project resources and it's also available at [20].

```
curl -H "Content-Type: application/json" \
-d '{"networkId":"fizzbuzz", "input":"aaaaaaaaa", "timeoutMillis":1000}' \
http://nepfix.herokuapp.com/compute

Output: ["a","fizz","aaaa","fizz","aa","aaaaaaa","fizz","buzz","aaaaaaaa"]
```

Fig. 4. Curl command for Γ execution

6 Conclusions and Future Work

From a computational viewpoint, We could conclude that this distributed implementation of NEPs neither imposes any additional constraint to the inherent capability of the computers involved in the simulation, nor adds any computational cost than the needed to the communication by means of the efficient HTTP protocol. We could draw, hence, that we propose an adequate and viable alternative to simulate NEPs in a distributed way. From a conceptual viewpoint, we could consider our approach as a kind of optimization to the general methodology for simulating natural computers described in section 2.2. As any optimization, it takes advantage of some specific characteristic of NEPs. The most obvious is that NEPs' topology is static, that is, it does not change while the NEPs evolve (that is not at all the general case). This is why we can simplify the communications module: each computer has a Java simulator for NEPs and a HTTP server to communicate to each other. By contrast, in the general case, we could need an independent broker to create more complex structures (like exchanges or queues, for example). Our immediate future work is related with two restrictions not addressed in this first version of Nepfix: 1) programmer has to explicitly design the physical topology (following the theoretical definition of NEP model) before starting the simulation and it can only be changed if cloud resources are accessed and are able to change it (it is possible the programmer

usually does not control the process) and 2) overcoming the limitation arising from proper resource management. The computations may be infinite (livelock) or blocked at some point because there is nothing to compute (deadlock). Finally, as part of our future work, NEP algorithms for NP complete problems must be designed and executed in Nepfix in order to prove their performance and their capacity for simulate them.

References

1. Alarcón, P., Arroyo, F., Mitrana, V.: Networks of Polarized Evolutionary Processors. Information Sciences **265**, 189–197 (2014)
2. Arroyo, F., Castellanos, J., Mitrana, V., Santos, E., Sempere, J.M.: Networks of Bio-inspired Processors, pp. 25–57. TRIANGLE, URV Publications (2012)
3. Arroyo, F., Gómez Canaval, S., Mitrana, V., Popescu, Ş.: Networks of polarized evolutionary processors are computationally complete. In: Dediu, A.-H., Martín-Vide, C., Sierra-Rodríguez, J.-L., Truthe, B. (eds.) LATA 2014. LNCS, vol. 8370, pp. 101–112. Springer, Heidelberg (2014)
4. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Solving NP-complete problems with networks of evolutionary processors. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2084, p. 621. Springer, Heidelberg (2001)
5. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Networks of Evolutionary Processors. Acta Informática. **39**, 517–529 (2003)
6. Díaz, M., Mingo, L., Gómez, N., Castellanos, J.: Implementation of Massive Parallel Networks of Evolutionary Processors (MPNEP): 3-Colorability Problem. Studies in Computational Intelligence **129**, 399–408 (2008)
7. Diez Dolinski, L., Núñez Hervás, R., Cruz Echeandía, M., Ortega, A.: Distributed simulation of P systems by means of map-reduce: first steps with hadoop and P-lingua. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part I. LNCS, vol. 6691, pp. 457–464. Springer, Heidelberg (2011)
8. García, M., Gutiérrez, R., Martínez, M., Orejuela, E., Pérez, I.: P-Lingua 2.0: A software framework for cell-like P systems. Int. Journal Computers, Communications and Control **IV**(3), 234–243 (2009)
9. Gómez, S., Batard, D., Gutiérrez, A.: A Web Implementation of A Generalized NEP. Int. Journal of Information Technologies and Knowledge **7**(1), 116–125 (2013)
10. Jiménez, A.: Simlación de Sistemas P distribuido en WAN. Escuela Politécnica Superior, Universidad Autónoma de Madrid, Tesis Máster (2013)
11. Lezzi, D., Rafanell, R., Torres, E., Giovanni, R., Blanquer, I., Badia, R.: Programming ecological niche modeling workflows in the cloud. Proc. IEEE Advanced Information Networking and Applications, 1223–1228 (2013)
12. Margenstern, M., Mitrana, V., Jesús Pérez-Jímenez, M.: Accepting hybrid networks of evolutionary processors. In: Ferretti, C., Mauri, G., Zandron, C. (eds.) DNA 2004. LNCS, vol. 3384, pp. 235–246. Springer, Heidelberg (2005)
13. Martínez, M., Macías, L., Valencia, L., Riscos, A., Pérez, M.: Accelerated Simulation of P Systems on the GPU: A Survey. Bio-Inspired Computing **472**, 308–312 (2014)
14. Manea, F., Margenstern, M., Mitrana, V., Pérez-Jiménez, M.J.: A New Characterization of NP, P, and PSPACE with Accepting Hybrid Networks of Evolutionary Processors. Theory Comput. Syst. **46**, 174–192 (2010)

15. Manea, F., Martín-Vide, C., Mitrana, V.: On the Size Complexity of Universal Accepting Hybrid Networks of Evolutionary Processors. Mathematical Structures in Computer Science **17**, 753–771 (2007)
16. Navarrete, C., Echeandia, M., Anguiano, E., Ortega, A., Rojas, J.: Parallel simulation of NEPs on clusters. Proc. of Int. Conf. of Web Intelligence and Intelligent Agent Technology, IEEE Computer. Society **3**, 171–174 (2011)
17. Navarrete, C.: Platform for Automatic Parallelization of Sequential Codes using Dynamic Graphs Partitioning and Based on User Adaptable Load Balancing. PhD. Thesis, Escuela Politécnica Superior, Universidad Autónoma de Madrid (2011)
18. Rees, J.: Fizz Buzz: 101 Spoken Numeracy Games. LDA Date (2002)
19. Pâun, G.: Computing with membranes. Journal of Computer and System Sciences **61**, 108–143 (1998)
20. Orgaz, P., Gómez S., Ortega, A.: Nepfix Project documentation (2015). https://github.com/pabloogc/Nepfix
21. Oasis.: Advanced Message Queuing Protocol. http://www.amqp.org/ (Accessed 12 February 2015)

How Nets of Evolutionary Processors (NEPs) Could be Simulated in a Distributed Way

Karina Jiménez[1(✉)], Antonio Jiménez[2], Marina de la Cruz[1], and Sandra Gómez Canaval[3]

[1] Departament of Computer Systems Engineering, Autónoma of Madrid University, Campus of Cantoblanco, 28049 Madrid, Spain
karina.jimenez@estudiante.uam.es, marina.cruz@uam.es
[2] FARA ASA, Trondheim, Norway
antonio.jimenez@fara.no
http://www.fara.no
[3] Department of Computer Systems, University College of Computer Science, Technical University of Madrid, Crta. de Valencia km. 7, 28031 Madrid, Spain
sgomez@etsisi.upm.es

Abstract. In this paper we describe a possible design to adapt to *Neps of Evolutionary Processors* (NEPs) a general methodogy for simulating natural computers in a distributed way. This methodology was early proposed by other researchers of our group and has proven to be viable and efficient for P-systems (another natural computer with a similar, and even more complex structure than NEPs). We highlight the strcuture, tasks and tools we plan to use in the future implementation of the system. Although several simulators for NEPs are available to the community via internet, almost none of them are designed to be scalable and able to tackle problems of big size.

Keywords: Networks of evolutionary processors · Natural computing · Distributed simulation · Simulation

1 Introduction

In this paper we face the distributed simulation of NEPs. It is organized as follows: we first introduce *Neps of Evolutionary Processors* (NEPs) and some state-of-the-art tools for distributed computing, then we introduce some approaches to the distributed simulation of natural computers, finally we describe our proposal that actually adatps a general methodology to the distributed simulation of natural computers that is being proposed by some researchers of our same group. In the last sections we summarize our conclusions and future research lines and list our references.

Results partially supported by Gain Dynamics, project iQuest which funding is applied to the Fondo Europeo de Desarrollo Regional para el Fomento de Sectores Tecnológicos de la Comunidad de Madrid, within the Estrategia Regional de Investigación e Innovación para una Especialización Inteligente (RIS3) in the Programa Operativo de la Comunidad de Madrid 2014-2020

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 55–68, 2015.
DOI: 10.1007/978-3-319-19258-1_5

1.1 NEPs

NEP stands for *Network of Evolutionary Processors*. NEPs are an abstract model of distributed/parallel symbolic processing presented in [2,3,13]. NEPs are inspired by biological cells. These are represented by words which describe their DNA sequences. Informally, at any moment of time, the evolutionary system is described by a collection of words, where each word represents one cell. Cells belong to species and their community evolves according to mutations and division which are defined by operations on words. Only those cells are accepted as surviving (correct) ones which are represented by a word in a given set of words, called the genotype space of the species. This feature parallels the natural process of evolution. Each node in the net is a very simple processor containing words which performs a few elementary tasks to alter the words, send and receive them to/from other processors. Despite the simplicity of each processor, the entire net can carry out very complex tasks efficiently. Many different works demonstrate the computational completeness of NEPs [5][11] and their ability to solve NP problems with linear or polynomial resources [10][3]. The emergence of such a computational power from very simple units acting in parallel is one of the main interests of NEPs.

NEPs can be used to accept families of languages. When they are used in this way they are called Accepting NEPs (ANEPs). Several variants of NEPs have been proposed in the scientific literature. NEP (the original model) [3], hybrid nets of evolutionary processor (HNEP) [5] and nets of splicing processors NEPS or NSP [11]. This last model uses a splicing processor, which adds a new operation (splicing rules) to mimic crossover in genetic systems. In section 3.1 we show an example of ANSP (the accepting variant of NSPs) solving the SAT problem. Nevertheless, all of them share the same general characteristics.

A NEP is built from the following elements:

a) A set of symbols which constitutes the alphabet of the words which are manipulated by the processors.
b) A set of processors.
c) An underlying graph where each vertex represents a processor and the edges determine which processors are connected so they can exchange words.
d) An initial configuration defining which words are in each processor at the beginning of the computation.
e) One or more stopping rules to halt the NEP.

An evolutionary processor has three main components:

a) A set of evolutionary rules to modify its words.
b) An input filter that specifies which words can be received from other processors.
c) An output filter that delimits which words can leave the processor to be sent to others.

The variants of NEPs mainly differ in their evolutionary rules and filters. They perform very simple operations, like altering the words by replacing all the occurrences of a symbol by another, or filtering those words whose alphabet is included in a given set of words.

NEP's computation alternates evolutionary and communication steps: an evolutionary step is always followed by a communication step and vice versa. Computation follows the following scheme: when the computation starts, every processor has a set of initial words. At first, an evolutionary step is performed: the rules in each processor modify the words in the same processor. Next, a communication step forces some words to leave their processors and also forces the processors to receive words from the net. The communication step depends on the constraints imposed by the connections and the output and input filters. The model assumes that an arbitrary number of copies of each word exists in the processors, therefore all the rules applicable to a word are actually applied, resulting in a new word for each rule. The NEP stops when one of the stopping conditions is met, for example, when the set of words in a specific processor (the output node of the net) is not empty. A detailed formal description of NEPs can be found in [2], [5] or [11]. For example, the following definition is literally taken from [2] and will help the reader to understand the model.

A network of evolutionary processors (NEP for short) of size n is a construct $\Gamma = (V, N_1, N_2, ..., N_n, G)$, where V is an alphabet and for each $1 \leq i \leq n, N_i = (M_i, A_i, PI_i, PO_i)$ is the i-th evolutionary node processor of the network. The parameters of every processor are:

- M_i is a finite set of evolution rules of one of the following forms only
  - $a \rightarrow b, a, b \in V$ (substitution rules),
  - $a \rightarrow \lambda, \in V$ (deletion rules),
  - $\lambda \rightarrow a, a \in V$ (insertion rules),

  More clearly, the set of evolution rules of any processor contains either substitution or deletion or insertion rules.
- A_i is a finite set of strings over V . The set A_i is the set of initial strings in the i-th node. Actually, we consider that each string appearing in any node at any step has an arbitrarily large number of copies in that node.
- PI_i and PO_i are subsets of V^* representing the input and the output filter, respectively. These filters are defined by the membership condition, namely a string $w \in V^*$ can pass the input filter (the output filter) if $w \in PI_i (w \in PO_i)$.

Finally, $G = (N_1, N_2, ..., N_n, E)$ is an undirected graph called the underlying graph of the network. The edges of G, that is the elements of E, are given in the form of sets of two nodes. The complete graph with n vertices is denoted by K_n.

1.2 Introduction to Some State-of-the-Art Tools for Distributed Computing

As we describe with more detail in further paragraphs, we propose to adapt a general methodology to compute any natural computer to NEPs. This methodology

tries to avoid a master-slave approach by explicilty taking the communications needs apart and using state-of-the-art tools to support them. Some of these tools (protobuff, the AMQP protocol and its rabbitMQ implementation) are introduced in the following points. Some of the explanations and examples are taken from the websites of these tools.

Protocol Buffers. Protocol buffers (https://code.google.com/p/protobuf/) were initially developed at Google to deal with an index server request/response protocol. They try to standarize the translation of data into a form that could be transmitted between computers. This process is usually named *serialization*.

Prior to protocol buffers, there was a format for requests and responses that used hand marshalling/unmarshalling, that is not a very desirable scenario.

In protobuff, adequate types of messages have to be defined to explicitly specify the way in which data are serialized. This is done by means of `.proto` files. Each protocol buffer message is a small logical record of information, containing a series of name-value pairs. Here's a very basic example of a `.proto` file that defines a message containing information about a person.

```
message Person {
  required string name = 1;
  required int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    required string number = 1;
    optional PhoneType type = 2 [default = HOME];
  }

  repeated PhoneNumber phone = 4;
}
```

There is a protocol buffer compiler that translates the `.proto` files into modules ready to be linked from the application being developed. For example the `Person.proto` file could be translated into a C++ `Person` class with methods to access, modify, parse and serialize data.

Protocol buffers are, thus, a flexible, efficient, automated mechanism for serializing structured data, similar to XML, but smaller, faster, and simpler.

AMPQ Protocol. AMQP (http://www.amqp.org/) is the internet protocol for bussines messaging, the Advanced Message Queuing Protocol (AMQP) is an open standard for passing business messages between applications or organizations. It connects systems, feeds business processes with the information they need and reliably transmits onward the instructions that achieve their goals. AMQP enables applications send and receive messages. In this regard it is like

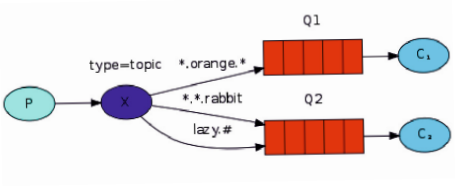
instant messaging or email. Where AMQP differs enormously is that it allows one to specify what messages will be received and from where, and how trade-offs are made with respect to security, reliability and performance.

RabbitMQ. RabbitMQ (http://www.rabbitmq.com/) is a messaging broker, that is, an intermediary for messaging. It implements AMQP and gives your applications a common platform to send and receive messages, and your messages a safe place to live until received.

Among the features provided by RabbitMQ we will focus on the *bindings-topics* mechanism. RabbitMQ extensibely uses queues to handle the messages. Topics refer to some kind of label you can attach to your messages to distinguish and classify them in some way. RabitMQ is able to link some queues to topics reresented by some restricted form of regular patterns. Topics are structured as a sequence of at most four words separated by the dot character (`.`) Messages are automatically delivered to those queues compatible with their topics. The basic interpretation of pattern in topics follows these rules:

- Characters different from `#` and `*` (hash and star) do not have any special meaning.
- The metacharacter `#` (hash), represents 0 or more words (including the separating dot character). For example, when a queue is bound to the pattern `#` (just this character), it will receive all the messages, like if you broadcast the message. This option (broadcast) is actually named *fanout exchange* in RabbitMQ. In the following paragraphs we explain other example.
- `*` is a metacharacter that represents any single word (that is, any sequence of any length excluding the separator)

We describe the *binding-topics* mechanism with more detail with the following example taken from (http://www.rabbitmq.com/): In this example, we're going to send messages about animals. The messages will be sent with a routing key that consists of three words: the first one describes speed, the second a colour, and the third a species, that is, `<speed>.<colour>.<species>`.



In this example we created three bindings: $Q1$ is bound to `*.orange.*`, $Q2$ to `*.*.rabbit` and `lazy.#`.

Their meaning can be summarised as:

- Q1 is interested in all the orange animals.
- Q2 wants to hear everything about rabbits, and everything about lazy animals.

1.3 Distributed Simulation of Natural Computers

There is currently almost none specific architecture to run neither bio-inspired models of computation, in general, nor NEPs in particular. They have to be simulated on traditional (von Neumann) computers.

The first simulator of NEP model named jNEP was introduced in [1,15]. jNep is a multithreaded Java simulator using massively parallel platforms for multicore desktop computers through Java RMI protocol. In this approach, the simulation is realized over the same hardware platform.

In [15,16] a framework able to parallelize any system that could be expressed as a set of logical nodes that communicate with others is introduced. It uses the MPI message passing interface and the C++ programming language. It follows an inherent master-slave structure. The simulation is performed in a synchronous way, that is, all the nodes compute the next step of the simulation at the same time, then they share their results and are ready for the next step. The load of each computer could be balanced between two simulation steps. There is a master processor in the cluster that is responsible of this task. The master, in addition, collects all the results of all the processors to send to each one the data needed for the next step. The master could also add or remove processors to the simulation in this moment. The master node has, hence, to be able to store the complete system after each simulation step. This is the main constrain to the size of the algorithm being simulated. Natural computers size often increases while their execution evolves.

An approach that tryes to overcome this drawback can be found in [9]. It proposes the use of specific tools of software industry to communicate parallel computations. The system structure could be summarized as folllows:

- AMQP (Advanced Messaging Queuing Protocol http://www.amqp.org/) is an industrial protocol to communicate computers in distributed environements. It is widely implemented, for example, by RabbitMQ (http://www.rabbitmq.com/). AMQP provides a set of mechanisms (like exchagnes, queues, bindings, topics, for example) able to tackle all typical interprocess communication needs in a general and efficient way that includes the possible access to distributed resources. In [9], the use of AMQP is proposed. It implies the definition of the corresponding communications protocol needed for the simulator. It has to include messages to share information between the subdomains run by each computer, but also messages to enroll and leave the simulation and to send the subdomain to new computers. A typical deployment of this approach includes three types of nodes:
  - Master, that should be named initial because this approach is not actually following the typical master-slave approach described in previous paragraphs. It is the first computer that starts the simulation.
  - Proxy's, or father nodes that has the power to split their domain to share it with a new computer that enrolls the simulation.
  - Slaves, that should be better named leaves, because are not able to split their domains any more.

– A complete simulator of the natural computer under consideration is needed. When run in distributed platforms this simulator will be responsible of just a subdomain of the complete system, but, if only one computer is running the system, it has to be able to do the whole work. The relationship of the AMQP broker with the computers is very important: each computer is supposed to have a copy of the simulator with a wrapper to handle the messages, but only master and proxy nodes need a copy of the broker running with the simulator.

Finally, we mentioned an important approach in membrane based computational models. There exists an important simulator using GPU's over conventional CPU's for them [12]. This approach is able to tackle big instances of the problems. In particular, the space constraints of simulators designed to be run on just a unique, even powerful, computer are avoided in this approach. This framework does not follow a general purpose approach. The researchers must write the code to simulate the specific instance of the model under consideration.

The main drawbacks in these parallel architectures described above are:

– The drawback of a general methodology that takes into account the difficulties discovered in previous approaches, to the distributed simulation of any natural computer. A general methodology implies, in our opinion, that new general purpose models of computation (like p-systems or NEPs, so on) become available to the researchers, as an alternative to classic von Neummann computers. In this sense, it is necessary to develop appropriate programming languages to describe how the problems are solved with these new computers.
– The space constraints of simulators designed to be run on just a unique, even powerful, computer. Real life problems are often too large to be tackled in this way.
– Once a researcher identifies the benefits from distributing a given algorithm, there is almost none direct way to access massive parallel hardware. One of the cheaper and more popular parallel architectures that our group have considered are clusters. We have explored different programming languages and approaches before discarding them:
  • Regarding Java, there is actually no standard JVM for clusters. There are, instead, several experimental approaches mainly based on the RMI protocol. These approaches usually don't focus on what we need to distribute the simulation of our systems among a set of processors.
  • Even with other programming languages better fitted for clusters (MPI message passing interface in the C++ programming language, for example) we have realized (in [16] and [15]) that clusters induce a master-slave approach and, hence, an important bottle neck that limits the size of the problem being solved: only small instances of interesting problems could be tackled before the system crashes due to lack of resources, and, hence, it can not be used to solve general real life problems.

The way that we propose to overcome these drawbacks are summarized as follows:

- To design a dynamic protocol for computers to join (and leave) at any moment the distributed simulation.
- With such a protocol we could avoid the bottle-neck associated with a master that has to be able to store the complete system.When a new computer enrolls the system, each computer only knows the part of the systems assigned to it.
- Given that distributing the execution of algorithms is a well known practice in industry, there already are well known, tested and efficient state-of-the-art tools to solve tasks such as the communications between computers.
  In the following paragraphs we introduce the modules that this approach considers necessary and some of the applications which implement these tools in industry. We have, in addition, previously checked other similar although less general approaches to the distributed simulation of membrane based systmes (P-systems). In [8] we have explored the possible use of the Map Reduce paradigm [6] by means of its implementation by Hadoop. We have concluded that, although it is possible and even could be efficient, Map Reduce is a natural way to store and query big amounts of data in a distributed way. It seems, hence, artificial to store P-systems into a distributed database and to translate the semantics of the P-system into a sequence of queries. We admit that it is a clever, but rather artificial, approach. This is why we have discarded it as the basis for a general methodology to simulate any natural computer.

2 Our Proposal

2.1 System Structure

The scheme of the system we have designed to be run by the computers that simulate our NEPs is shown in Figure 1

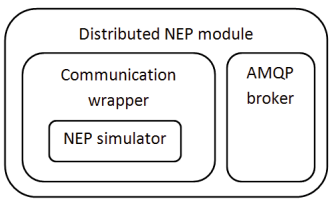


Fig. 1. Stgructure of the system run by each computer

It includes the following modules:

- The NEP simulator itself. We have to code the behaviour of NEPs.
- In the communication step, the simulator has to allow both options to communicate the processors: local and remote nodes. This is accomplished by means of a "communications wrapper" that hides these details. Processors will write and read to and from their connections (that could be instances of the proper classes in an object oriented implementation) wihtout actually knowing if these connections are local (inside the same computer) or remote (the processors are run by different computers).
- Each computer has potentially to be running the AMQP broker selected (we plan to use rabbitMQ)
- An adequate communication protocol has to be defined (and implemented by means of the broker resources). We need the following kinds of messages:
  - Messages for the distributed simulation of the NEP. This kind of messages includes both, those to simualte the communication step, and those to implement a mechanism to incorporate a new computer to the simulation.
  - Messages to synchronize the NEP. In a local simulation, synchronizing the processors is a strightforward task. Nevertheless, in a distributed and concurrent environment, synchronization becomes a non trivial issue.

In the following sections we will describe both.

2.2 Messages Needed for the Distributed Simulation of the NEP

We will introduce our proposal by means of a simple example from which it is easy to generalize our approach. Let us consider the NEP whose graphical scheme is showin in figure 2. It includes 5 processors $\{n_1, n_2, n_3, n_4, n_5\}$. Its connections are $\{(n_1, n_3)(n_1, n_4)(n_2, n_3)(n_2, n_4)(n_4, n_5)\}$.

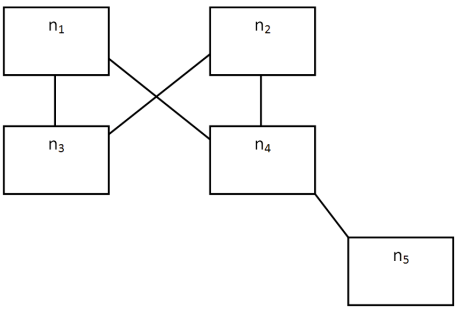


Fig. 2. Topology of the NEP example

Remember that it is an undirected graph. We only will phocus on the system architecture not in the semantics of this toy example. This is why we do nont

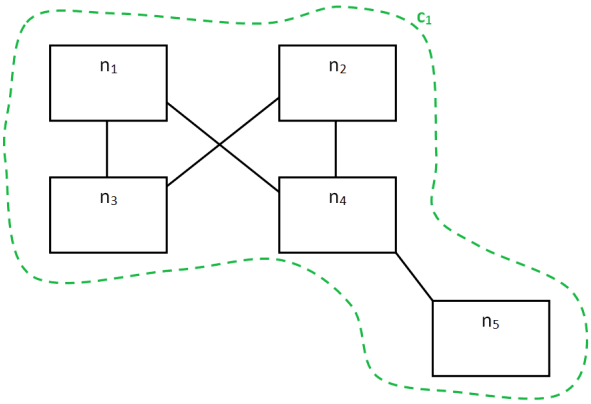


Fig. 3. Initially, c_1 is the only computer that runs the NEP

mention neither any content nor rule. One of the most important characteristics of our approach is the possibility of dinamically assigning some part of the systems to be computed by other computers that joins the task of executing the NEP. Let us describe how the system works:

1. There will initially be only one computer (c_1) to run all the NEP as figure 3 shows.
2. After a while, a new computer c_2 sends a message to c_1 asking to join the simulation. Computer c_1 handles this message as follows:
   - It runs the algorithm to split the domain. In our example the resulting subgraphs are $G_1 = \{\{n_1, n_2, n_3\}, \{(n_1, n_3), (n_2, n_3)\}\}$ and $G_2 = \{\{n_4, n_5\}, \{(n_4, n_5)\}\}$. Notice that nodes n_1, n_2 and n_4 become *edge nodes* that are responsible for communicating both computers. In this way, the original connections (n_1, n_4) and (n_2, n_4) actually communicate computers c_1 and c_2.
   - There are different alternative structures to implement the *logical* architecture into the *physical* net of computers. As we have previously described we propose to use the standard AMQP (Advanced Message Queuing Protocol). We plan to use its rabbitMQ implementation. We will use, in particular
     - A rabbitMQ *exchange* to communicate the NEP's processors.
     - A set of rabbitMQ *queues* (one for each edge node)
     - We propose to implement our communication protocol by means of the rabbitMQ *bindings and topics* mechanism, by means of which, each queue is bound to a set of topics. Whenever a computer writes to the exchange, its messages are copied to all the queues compatible with the messages' topics. Processes have just to read from the proper queue to get their messages.
   - A possible solution could create the communication structure graphically schetched in figure 4 taking the following steps:

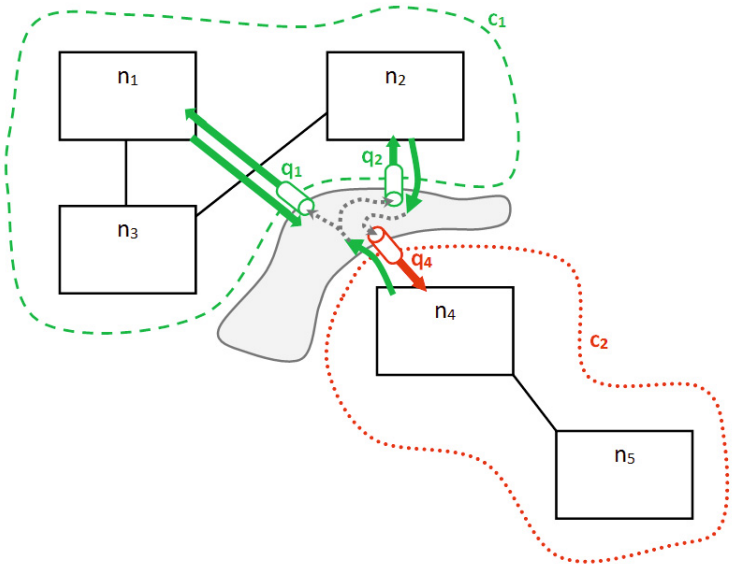


Fig. 4. Communication structure after processing the message from c_2 asking for joining the simulation

(a) First, the computer c_1 (as any other node that receives a message asking for joinning the simulation) creates a new queue for each of its edge nodes (in this case queues q_1 and q_2 respectively for nodes n_1 and n_2).
(b) Then, it (c_1) sends to c_2 the list of queues (one for each of its edge nodes) and topics that c_2 should create and bind to. In our example c_2 has to create a queue for the node n_4 (q_4) and bind it to topics that indicate that only messages from node n_1 and n_2 are welcome (for example `*from_nep1*` and `*from_nep2*`).
(c) And finally, c_1 sends to c_2 its subgraph (G_2, in this case).

3. NEP's simulation continues from this point. Notice that
   – In the *evolution step* the NEP simulator module computes all the resulting strings of its nodes. Each node finds its inputs from the net no matter if the other nodes are local or remote. This is accomplished by the *communication wrapper* that will read from either local nodes or from rabbitMP queues in the case of remote nodes.
   – In the *communication step* each node writes its outputs in the net. The *commnication wrapper* (once again) hides the differences between writting in a local node of in the exchange (for remote nodes). In the most general case, each edge node, therefore, reads its messages from its queue, and writes its messages to the exchange.

2.3 Messages Needed for Synchronizing the Processors

Synchronization of distributed processing nodes like these is a classic problem for which there are classic solutions. We describe in this section and rather *ideal* and general scenario. We could summarize the synchronization of our distributed simulation of NEPs in the following way:

- A new *virtual* node (*clock*) is added to the net.
- Its goal could be described as follows:
  1. To collect messages from the other nodes when they have finished their *evolution step*.
  2. Then, to send to the other nodes messages to advertise them they can continue computing.
- And, hence, the NEP simulator run on each computer has to perform the next *evolution step* only when it receives a message allowing it; and has to send to the clock a message saying *"My output is ready"* after finishing the evolution step.
- The NEP simulator only can perform an *evolution step* when all its nodes receive from *clock* the corresponding message.

Several options are possible to synchronize the simulation. In the worse case we could assume that each node neads a queue to read the message to start computing its inputs. This could be the case if each node is assigned to a different computer. Nodes could write to a syncrhonizing AMQP exchange their messages informing that their outputs are ready. The *clock* could *broadcast* the *continue computation* message to all the processors by means of, for example, the rabbitMQ `fanout` type. The processors could use the bindings-topics mechanism (for example with the topic `*node_i_ready*`).

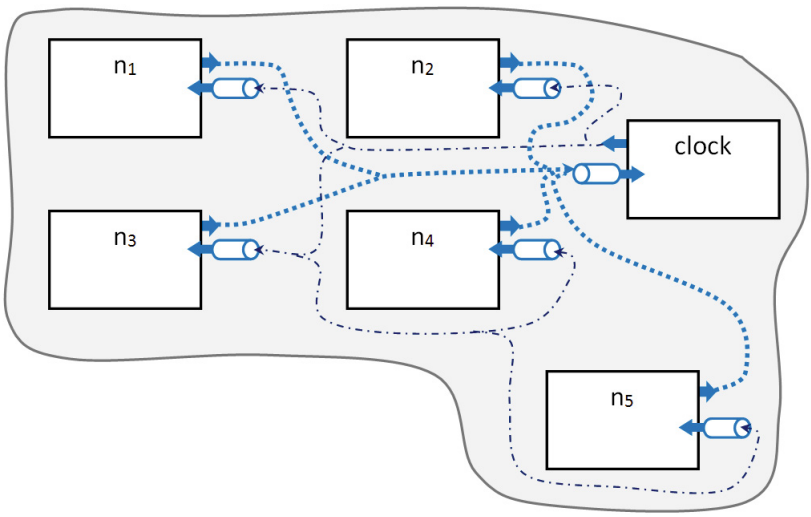


Fig. 5. A possible synchronization structure

Figure 5 shows a possible architecture for synchronizing the simulation. To simplify it we have removed any reference to computers. The shaded cloud represents the common presence of the broker wherever it is necessary. This scheme is, hence, a simplified version. In a more real scenario a configuration more similar to that shown in figure 4 will be needed.

3 Conclusions and Further Research Lines

We have shown, by means of a simple example, how to adapt to NEPs a new general methodology to simulate any natural computer that is being proposed by our research group.

It is important to highlight that we are actually using this approach in a complete implementation that is not ready at this moment.

We describe an structure that could be scaled to general cases and tackle the main problems of the distributed simulation of NEPs: the synchronization of processors, their distributed and concurrent computation and the usual presence of bottle-necks when massively parallel resources are accessed. The main advantage of such a simulator is the possibility of tackling real life problems that usually have a size too large for being solved by simple personal computers.

In the future we plan to complete the implementation of our approach and to compare its performance with other state-of-the-art approaches. Several lines of our group strongly depends on a simulator like this to perform their experiments.

We plan to offer the comunity a general purpose NEPs simulator able to run any kind of NEP. It will be integrated in a complete developing environment that makes this model useful for solving problems in different domains. This platform is being designed by several researchers of our group.

References

1. Arroyo, F., Castellanos, J., Mitrana, V., Santos, E., Sempere, J.M.: Networks of Bio-inspired Processors, Triangle, pp. 25–57. URV Publications (2012)
2. Castellanos, J., Martin-Vide, C., Mitrana, V., Sempere, J.M.: Networks of evolutionary processors. Acta Informatica **39**(6–7), 517–529 (2003)
3. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Solving NP-complete problems with networks of evolutionary processors. In: Mira, J., Prieto, A.G. (eds.) IWANN 2001. LNCS, vol. 2084, p. 621. Springer, Heidelberg (2001)
4. Choudhary, A., Krithivasan, K.: Network of evolutionary processors with splicing rules. Proceedings of the Mechanisms, Symbols and Models Underlying Cognition, PT 1 **3561**, 290–299 (2005)
5. Csuhaj-Varju, E., Martin-Vide, C., Mitrana, V.: Hybrid networks of evolutionary processors are computationally complete. Acta Informatica **41**(4–5), 257–272 (2005)
6. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)

7. Diaz, M.A., Gomez Blas, N., Santos Menendez, E., Gonzalo, R., Gisbert, F.: Networks of evolutionary processors (nep) as decision support systems. In: Fifth International Conference on Information Research and Applications, vol. 1, pp. 192-203. ETHIA (2007)
8. Diez Dolinski, L., Núñez Hervás, R., Cruz Echeandía, M., Ortega, A.: Distributed simulation of P systems by means of map-reduce: first steps with Hadoop and P-Lingua. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part I. LNCS, vol. 6691, pp. 457–464. Springer, Heidelberg (2011)
9. Jiménez, A.: Simlación de Sistemas P distribuido en WAN, Tesis de Máster. http://www.eps.uam.es/nueva_web/intranet/ga/tfdm/trabajos/Antonio_Jimenez_Martinez.pdf
10. Manea, F., Martín-Vide, C., Mitrana, V.: All NP-problems can be solved in polynomial time by accepting networks of splicing processors of constant size. In: Mao, C., Yokomori, T. (eds.) DNA12. LNCS, vol. 4287, pp. 47–57. Springer, Heidelberg (2006)
11. Manea, F., Martin-Vide, C., Mitrana, V.: Accepting networks of splicing processors: Complexity results. Theoretical Computer Science **371**(1–2), 72–82 (2007)
12. Martínez-del-Amor, M.A., Macías-Ramos, L.F., Valencia-Cabrera, L., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Accelerated simulation of P systems on the GPU: a survey. In: Pan, L., Păun, G., Pérez-Jiménez, M.J., Song, T. (eds.) BIC-TA 2014. CCIS, vol. 472, pp. 308–312. Springer, Heidelberg (2014)
13. Martin-Vide, C., Mitrana, V., Perez-Jimenez, M.J., Sancho-Caparrini, F.: Hybrid networks of evolutionary processors. Proceedings of the Genetic and Evolutionary Computation. GECCO 2003, PT I **2723**, 401–412 (2003)
14. Martin-Vide, C., Mitrana, V.: Solving 3CNF-SAT and HPP in linear time using WWW. Machines, Computations, and Universality **3354**, 269–280 (2005)
15. Navarrete, C., Echeandia, M., Anguiano, E., Ortega, A., Rojas, J.: Parallel simulation of NEPs on clusters. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 171–174. IEEE Computer Society (2011)
16. Navarrete, C.: Platform for automatic parallelization of sequential codes using dynamic graphs partitioning and based on user adaptable load balancing, PhD. thesis. https://repositorio.uam.es/handle/10486/7514

Brain-Computer Interfaces: Applications and Tele-services

A Comparison of SSVEP-Based BCI-Performance Between Different Age Groups

Felix Gembler, Piotr Stawicki, and Ivan Volosyak(✉)

Faculty of Technology and Bionics, Rhine-Waal University of Applied Sciences, Kleve, Germany
ivan.volosyak@hochschule-rhein-waal.de
http://www.hochschule-rhein-waal.de

Abstract. In this paper we compare the performance of a SSVEP-based BCI spelling application of two different equally sized age groups (five subjects each, ranging from 19 to 27 years and 66 to 70 years). Our results confirm that elderly people may have a slightly deteriorated information transfer rate (ITR). The mean (SD) ITR of the young age group was 27.18 (8.82) bit/min while the elderly people achieved an ITR of 14.42 (6.29) bit/min. The results show that the subject age must be taken into account during the development of a SSVEP-based application.

Keywords: BCI (Brain-Computer Interface) · SSVEP (Steady-State Visual Evoked Potential) · LCD (Liquid Crystal Display) · Speller · Age

1 Introduction

A brain-computer interface (BCI) is a technical system that acquires and analyses brain activity patterns in real time to translate them into control commands for computers or external devices [4,14]. Steady-state visual evoked potentials (SSVEP) are the continuous brain responses elicited at the occipital and parietal cortical areas under visual stimulation with a specific constant frequency; they can be used for BCI and are described as reliable in relevant literature [9]. Presently, the SSVEP approach provides the fastest and most reliable communication paradigm for the implementation of a non-invasive BCI system. BCI-applications can assist people paralyzed by disorders such as cerebral palsy, spinal cord injury, brain stem stroke, amyotrophic lateral sclerosis (ALS), or muscular dystrophies to participate in daily life activities [13]. Those disorders can be found among all age groups. Also, the effects of aging alone present physical limitations that all-too-often prevent older people from interacting with their environment. Still, even though handicapped people from all age groups could benefit from BCI-Technologies, the majority of BCI systems were tested using younger subjects. Some articles have previously reported slightly worse BCI performance by older subjects. E.g., in a 12 participant study about latency and distribution of P300 (another commonly used BCI-approach using the 300ms component of an evoked potential), N.S. Dias et al. found that older subjects (>51 years) show

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 71–77, 2015.
DOI: 10.1007/978-3-319-19258-1_6

smaller P300 amplitudes than younger ones [2]. Grosse-Wentrup and Schölkopf reviewed performance variations in BCIs based on the sensorimotor-rhythm (SMR) and stated that a correlation between age and BCI performance is conceivable [6]. Furthermore, Macpherson et al. investigated age-associated changes in SSVEP amplitude and latency with memory performance [8]. They found that older adults demonstrated reduced neural activity during lower task demands, whereas with greater task demands, their neural activity was increased. Research on accuracy in SSVEP based BCIs frequently reported variations in performance between users. Ehlers et al. reported age group distinctions concerning accuracy rates of a performance with a SSVEP-based spelling application [3], but only children and young adults between 6 and 33 where tested here. The young adults obtained higher accuracy rates compared to children. Research articles on so-called BCI demographics in SSVEP-BCIs also reported age-related performance differences. In a larger study, the spelling performance with a SSVEP-based spelling application was analysed and it was observed that younger subjects were less annoyed by the flickering and tended to attain a higher information transfer rate (ITR) [1]. However in this study few subjects were over 50 years old. In another subsequent demographics study, subjects between 18 to 55 years were tested, but neither an effect on age, gender, nor there interaction were statistically significant [10]. In order to explore this performance difference further, we tested two equal sized groups of different age ranges with a SSVEP-spelling application. High ITRs are an essential goal for any BCI. The arrangement of the visual stimuli and the desired targets is another key factor in ensuring effective control. Especially for older people, the readability and simplicity of the graphical user interface (GUI) are crucial. Because of this, we used a rather small number of simultaneously displayed targets to reduce cognitive load. As opposed to five classes as in [1,3,12], the spelling application used for this experiment displayed only four boxes containing the alphabet which flickered simultaneously.

The paper is organized as follows: the second section describes the experimental setup, and presents details about the spelling interface. The results are presented in the third section, followed by a discussion and conclusion in the final section.

2 Methods and Materials

2.1 Subjects

Two groups of five healthy volunteer subjects each participated in the study. The group of younger subjects (*groupA*) had a mean (SD) age of 22.6 (2.88), ranging from 19 to 27. All subjects from this group were students or employees of the Rhine-Waal University of Applied Sciences and had little or no previous experience with BCI systems. Four subjects of this group were female. The other group (*groupB*) consisted of two male and three female subjects, with a mean (SD) age of 68 (2), ranging from 66 to 70. No subjects from *groupB* had ever used a BCI, and did not report any mental or physical disability.

The EEG recording took place in a standard laboratory room with low background noise and luminance. None of the subjects had neurological or visual disorders. Spectacles were worn when appropriate. All persons who volunteered to participate in the study became research subjects after reading the subject information sheet and signing a consent form. The subjects did not receive any financial reward for participating in this study.

2.2 Signal Acquisition

Subjects were seated in front of a LCD screen (BenQ XL2420T, resolution: 1920×1080 pixels, vertical refresh rate: 120 Hz) at a distance of about 60 cm. The used computer system operated on Microsoft Windows 7 Enterprise running on an Intel processor (Intel Core i7, 3.40 GHz). Standard Ag/AgCl electrodes were used to acquire the signals from the surface of the scalp. The ground electrode was placed over AF_Z, the reference electrode over C_Z, and the eight signal electrodes were placed at predefined locations on the EEG-cap marked with $P_Z, PO_3, PO_4, O_1, O_2, O_Z, O_9$ and O_{10} in accordance with the international system of EEG electrode placement. Standard abrasive electrolytic electrode gel was applied between the electrodes and the scalp to bring impedances below 5 $k\Omega$. An EEG amplifier, g.USBamp (Guger Technologies, Graz, Austria), was utilized. The sampling frequency was set to 128 Hz. During the EEG signal acquisition, an analogue band pass filter (between 2 and 30 Hz) and a notch filter (around 50 Hz) were applied directly in the amplifier.

2.3 SSVEP-Based Three-Step Spelling Application

The *Three-step spelling application* resembles an earlier developed GUI layout [5,7,12]. In the *Three-step spelling application* four commands were represented on the computer screen by flickering boxes of default sizes (175 x 175 pixels). The size of the boxes varied during the experiment as described in [11]. The subject faced four boxes and in order to increase user friendliness, the user commands were displayed in the subjects mother tongue (in all cases German). Three boxes were arranged horizontally in the upper part of the screen containing the letters "A-I", "J-R" and "S-_", respectively. The additional 4th box, containing the command "Löschen" (delete the last spelled character) was located on the right side of the screen. The box for the written word and the word to spell was placed in the centre of the screen. The content of the three boxes containing the alphabet changed to more specific sets according to the first selection made. The boxes would then display "A B C", "D E F", "G H I" or "J K L", "M N O", "P Q R" or "S T U", "V W X", "Y Z _". After selection in this second window, the content of the boxes changed once more, and each box contained a single letter. In both the second and the third window, the far right box ("Löschen" in window 1) would contain the command "zurück" (back), giving the user the option to switch to the previous window. At least three steps were necessary to choose any single letter. If the subject made a mistake, and corrected it with the command "zurück" (back), the number of steps would increase. A screenshot

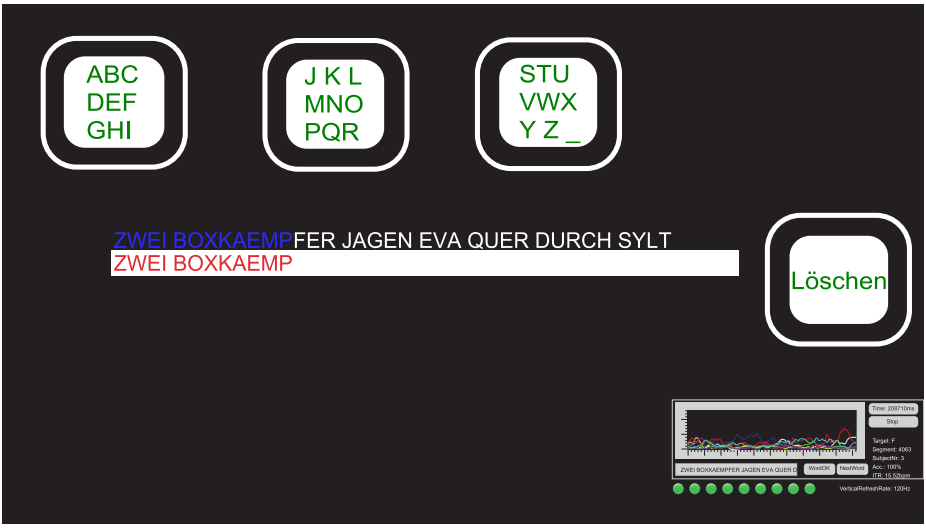


Fig. 1. GUI of the *Three-step spelling application* during the online experiment. A subject was spelling the text "ZWEI BOXKAEMPFER JAGEN EVA QUER DURCH SYLT" (a pangram).

of the first window taken during the online spelling task is shown in Figure 1. In order to reduce the information load of the visual channel, every command classification was followed by an audio feedback with the name of the selected command or the letter spelled (also in German). The SSVEP classification was performed online every 13 samples (ca. 100ms) on the basis of the adaptive time segment length of the acquired EEG data. More details about the used SSVEP detection method can be found in [11]. In order to make the system more robust we included larger time windows (8 and 16 seconds) and increased the number of time segment lengths, which are shown in Figure 2.

2.4 Experimental Setup

After signing the consent form, each subject completed a brief questionnaire and was prepared for the EEG recording. Subjects participated in a familiarization run spelling the word "KLEVE", and a word of choice (e.g. the own first name). Next, each subject used the GUI to spell the German phrase "ZWEI BOXKAEMPFER JAGEN EVA QUER DURCH SYLT". Stimulation frequencies and other SSVEP-key parameters that were used in this experiment were determined individually on the basis of the refresh rate of the LCD screen (120 Hz) during the familiarization run. Frequencies between 6.67 Hz and 12.00 Hz were used. Each spelling phase ended automatically when the presented word was spelled correctly. Spelling errors were corrected via the implemented "Löschen" button. Information needed for the analysis of the test was stored anonymously during the experiment. The entire session took on average

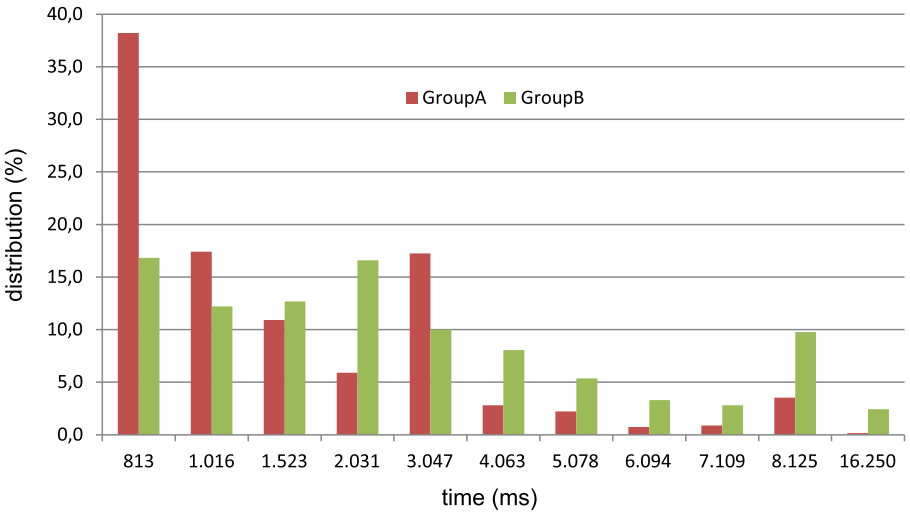


Fig. 2. Distribution of time segment lengths for each age group

about 60 minutes for each subject. All data collected during the experiment were recorded anonymously.

3 Results

BCI performance for each subject was evaluated by calculating the commonly used ITR in bit/min, employing the formula as discussed e.g. in [14]. In the GUI presented here, the overall number of possible choices was four. The accuracy was calculated based on the number of correct command classifications divided by the total number of classified commands. The overall BCI performance is given in Table 1. All ten subjects were able to complete the spelling phrase. The distribution of time windows is displayed in Figure 2.

4 Discussion and Conclusion

It can be seen in Table 1 that there is a substantial difference between the performance of younger and older subjects. Subjects from *groupA* reached a mean accuracy of 97.29 %. Even two subjects from this group completed the spelling task without errors, achieving an accuracy of 100 %. The mean accuracy of *groupB* was 89.12 % and no subject of this group reached 100 % accuracy. Also the time needed to complete the spelling task was noticeably larger for subjects from *groupB*. The mean ITR of *groupA* was 27.18 bit/min while subjects from *groupB* achieved an ITR of only 14.42 bit/min. For subjects from *groupB* the classification time window was usually larger (see Figure 2). Explanations for

Table 1. Results for both age groups. Mean values are given at the bottom of the table

	Group A				**Group B**		
	Age 19-27				Age 66-70		
Subject	Time sec	Acc %	ITR bpm	Subject	Time sec	Acc %	ITR bpm
1	466.27	98.48	31.62	6	1841.23	79.65	6.90
2	468.00	100.00	32.31	7	1097,84	90.38	11.84
3	409.20	100.00	36.95	8	963.82	84.94	11.85
4	1114.14	91.55	11.99	9	815.14	92.96	15.87
5	627.05	96.43	23.03	10	548.54	97.69	25.65
Mean	**616.93**	**97.29**	**27.18**	**Mean**	**1053.31**	**89.12**	**14.42**
SD	258.98	3.15	8.82	SD	434.13	6.28	6.29
Min	**409.20**	**91.55**	**11.99**	**Min**	**548.54**	**79.65**	**6.90**
Max	**1114.14**	**100.00**	**36.95**	**Max**	**1841.23**	**97.69**	**25.65**

poorer performance might be that younger subjects had shorter reaction time and also faster learning ability compared to the older subjects. It should also be noted that the performance gap could be even larger if a higher number of stimulation targets would be displayed, as the elderly people might have more problems with an increased information load of the visual channel. Future work should address the performance gap caused by advanced age. GUIs could be modified to suit the needs of older users.

Acknowledgments. This research was supported by the German Federal Ministry of Education and Research (BMBF) under Grant 16SV6364. We also thank all the participants of this research study as well as our student assistants Catharina Thoma and Julia Falkenstein.

References

1. Allison, B., Lüth, T., Valbuena, D., Teymourian, A., Volosyak, I., Gräser, A.: BCI Demographics: How many (and what kinds of) people can use an SSVEP BCI? IEEE Trans. Neural Syst. Rehabil. Eng. **18**(2), 107–116 (2010)
2. Dias, N., Mendes, P., Correia, J.: Subject age in P300 BCI. In: Proceedings of the 2 International IEEE EMBS, March 2005
3. Ehlers, J., Valbuena, D., Stiller, A., Gräser, A.: Age-specific mechanisms in an SSVEP-based BCI scenario: Evidences from spontaneous rhythms and neuronal oscillators. Computational Intelligence and Neuroscience 2012, 20 (2012)
4. Gao, S., Wang, Y., Gao, X., Hong, B.: Visual and auditory brain-computer interfaces. IEEE Transactions on Biomedical Engineering **61**(5), 1436–1447 (2014)
5. Gembler, F., Stawicki, P., Volosyak, I.: Towards a user-friendly BCI for elderly people. In: Proceedings of the 6th International Brain-Computer Interface Conference Graz (2014)
6. Grosse-Wentrup, M., Schölkopf, B.: A review of performance variations in SMR-based brain- computer interfaces (BCIs). In: Guger, C., Allison, B.Z., Edlinger, G. (eds.) Brain-Computer Interface Research, pp. 39–51. Springer, Heidelberg (2013)

7. Kick, C., Volosyak, I.: Evaluation of different spelling layouts for SSVEP based BCIs. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1634–1637. IEEE (2014)
8. Macpherson, H., Pipingas, A., Silberstein, R.: A steady state visually evoked potential investigation of memory and ageing. Brain and Cognition **69**, 571–579 (2009)
9. Vialatte, F.B., Maurice, M., Dauwels, J., Cichocki, A.: Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. Prog. Neurobiol. **90**, 418–438 (2010)
10. Volosyak, I., Valbuena, D., Lüth, T., Malechka, T., Gräser, A.: BCI Demographics II: How many (and what kinds of) people can use an SSVEP BCI? IEEE Trans. Neural Syst. Rehabil. Eng. **19**(3), 232–239 (2011)
11. Volosyak, I.: SSVEP-based Bremen-BCI interface - boosting information transfer rates. J. Neural Eng. **8**(3), 036020 (2011)
12. Volosyak, I., Moor, A., Gräser, A.: A dictionary-driven SSVEP speller with a modified graphical user interface. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part I. LNCS, vol. 6691, pp. 353–361. Springer, Heidelberg (2011)
13. Wolpaw, J.R.: Brain-Computer Interface. Encyclopedia of Neuroscience **1**, 429–437 (2009)
14. Wolpaw, J., Birbaumer, N., McFarland, D., Pfurtscheller, G., Vaughan, T.: Brain-computer interfaces for communication and control. Clin. Neurophysiol. **113**, 767–791 (2002)

Training in Realistic Virtual Environments: Impact on User Performance in a Motor Imagery-Based Brain–Computer Interface

Leandro da Silva-Sauer[1], Luis Valero-Aguayo[2], Francisco Velasco-Álvarez[1(✉)], Sergio Varona-Moya[1], and Ricardo Ron-Angevin[1]

[1] Departamento de Tecnología Electrónica, Universidad de Málaga, Málaga, Spain
{sauer,fvelasco,sergio.varona,rron}@uma.es
[2] Departamento de Personalidad, Universidad de Málaga,
Evaluación y Tratamiento Psicológico, Málaga, Spain
lvalero@uma.es

Abstract. A brain–computer interface (BCI) is a system that enables people to control an external device by means of their brain activity, without the need of performing muscular activity. BCI systems are normally first tested on a controlled environment before being used in a real, daily scenario. While this is due to security reasons, the conditions that BCI systems users will eventually face in their usual environment may affect their performance in an unforeseen way. In this paper, we try to bridge this gap by presenting a trained BCI user a virtual environment that includes realistic distracting stimuli and testing whether the complexity or the type of such stimuli affects user performance. 11 subjects navigated two virtual environments: a static park and the same one with visual and auditory stimuli simulating typical distractors from a real park. No significant differences were found when using a realistic environment; in other words, the presence of different distracting stimuli did not worsen user performance.

Keywords: Brain-Computer Interface (BCI) · Virtual Environment (VE) · Distraction · Visual stimuli · Auditory stimuli

1 Introduction

A brain–computer interface (BCI) is a system that enables a special type of communication, one that is not based on muscular movements but rather on brain activity alone. This activity can be non-invasively recorded by means of scalp electrodes, thus obtaining the electroencephalogram (EEG).

Some of these systems rely on the brain activity that is elicited when a person merely imagines himself/herself moving a part of the body, which is similar to the activity elicited by actually performing the movement [1]. Such systems, called sensorimotor rhythm BCI (SMR-BCI), analyze specific frequency components of the EEG that change with MI. The most frequently used component is the μ rhythm, whose frequency band is located approximately between 8 and 13 Hz. When a motor brain area is not involved in a specific motor task, certain synchronized activity can be

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 78–88, 2015.
DOI: 10.1007/978-3-319-19258-1_7

detected as a power increase in the μ band. However, when this same area is engaged in a motor activity—both real and imagined—, a desynchronization of the neural activity usually happens (event-related desynchronization, ERD), which is observed as a power decrease of the μ rhythm amplitude. Therefore, these detectable EEG changes can be translated in real time to specific commands for different purposes in the field of medicine and particularly neurorehabilitation [2-4].

However, the use of SMR-BCIs is presently limited by their relative low reliability, specially when compared when other BCIs, such as those based on event-related potentials like the P300. Unlike the latter, SMR-BCIs demand that users undergo a preliminary training protocol (see [5] for a review) in order to achieve a sufficient control of the system. The desired degree of control depends, in fact, of the environment on which the BCI will eventually be used. This is particularly important in the case of BCI systems for navigation applications, since driving a wheelchair through one's home is remarkably different from driving it along a promenade or inside a park. In the latter cases it is necessary to guarantee that the user has enough control to avoid the dangerous situations that will almost inevitably arise.

In this regard, the use of a virtual environment (VE) as part of the training protocol is a most suitable choice for researchers to estimate how user performance with a BCI would be affected by distracting situations from a real life scenario. Apart from the security issue, using a VE has the advantage of its realism and the intensity of the feedback received by the user [6, 7], so he/she may face a controlled but distracting situation that resembles a real one.

In this context, the objective of the present study was to verify the effect of visual and auditory distracting stimuli (i.e., stimuli that may be present in ordinary life situations) on user performance with a MI-based BCI for controlling a navigation application. For this we followed a paradigm that was used in a previous study [8] to compare user performance when navigating a mobile robot on a real environment with the performance on a virtual environment that closely resembled the former.

2 Methods

2.1 Participants

11 naïve voluntary students (nine female, $M = 21.81$ years, $SD = 1.77$ years) from the Faculty of Psychology participated in this study. They were selected from a greater sample of volunteers according to the criterion of having a minimum error rate lower than 30% in a calibration phase [9]. They were informed of the research aim, the technology involved, the duration and the possible effects of the use of virtual reality (VR). All of them signed an informed consent and received 0.5 bonus points in the grade of a specific subject.

2.2 Procedure

The study was carried out on three different days or *sessions*. A number of phases, whose characteristics are shown in Table 1, were divided among these sessions.

The first session consisted only of carrying out phase A, which consisted of an initial training phase for calibration purposes and a subsequent adaptation phase. We followed the paradigm proposed by our group (UMA-BCI) in [7], which, in turn, was based on the paradigm proposed by the Graz group [10]. In our paradigm, the participants were immersed in a VE that resembled a three-lane country highway. A car was moving along the central lane. The participants were asked to perform right hand MI or to stay in a mentally relaxed state if a water puddle appeared in the right or in the left lane, respectively. When receiving feedback, those actions would move the car to the right or to the left, respectively. The calibration phase took approximately 30 minutes, excluding the time needed to set up the EEG recording apparatus.

If the participant's EEG data could be classified by a linear discriminant analysis (LDA) classifier with a minimum estimated error below 30%, then the participant proceeded with the adaptation phase, in which he/she was allowed to freely use the navigation application proposed in [8] to navigate through a VE in order to get accustomed to its visual and auditory interface. This part took 5–10 minutes.

In sessions two and three the participants were asked to complete a navigation task with the visual and auditory interface and with the auditory interface alone, respectively. In both sessions participants carried out the task with and without the presence of distracting stimuli. To prevent carryover effects we counterbalanced the delivery of those stimuli. Thus, six participants carried out the tasks according to the sequence B1-B2-C2-C1, whereas the remaining five followed the sequence B2-B1-C1-C2. Therefore, all the participants went through every phase. These two sessions took approximately one hour.

Table 1. Characteristics of the different phases carried out in the study

Name	Purpose	Application interface	Distracting stimuli
A	Calibration and adaptation		
B1	Evaluation	Visual + Auditory	No
B2	Evaluation	Visual + Auditory	Yes
C1	Evaluation	Auditory	No
C2	Evaluation	Auditory	Yes

The navigation task from the evaluation phases (i.e., phases B1, B2, C1, and C2) consisted of navigating through a virtual park by means of making 21 consecutive selections of movement commands. Each selection was considered a trial. While the virtual park in phases B1 and C1 consisted only of static visual stimuli (i.e., the elements of the park), the one in phases B2 and C2 had also a variety of dynamic visual and auditory stimuli.

2.3 Apparatus and Materials

The experiments were carried out in a soundproof 4 m x 2.4 m room. Participants seated 3 m in front of a stereoscopic overhead projector screen (1.7 m x 2.3 m), wearing polarized glasses. Auditory stimuli were conveyed by means of a 5.1 surround sound system.

The EEG was recorded from two bipolar channels. The active electrodes were placed 2.5 cm anterior and posterior to electrode positions C3 and C4 (right and left hand sensorimotor areas, respectively) according to the 10/20 international system. The ground electrode was placed at the FPz position. Signals were amplified by a 16 channel biosignal g.BSamp (Guger Technologies) amplifier and then digitized at 128 Hz by a 12-bit resolution data acquisition NI USB-6210 (National Instruments) card. The offline and online EEG data processing was done with MATLAB 2013a. All statistical analyses were done using SPSS.

2.4 Signal Processing

EEG processing and feedback generation was based on the procedure detailed in [11]. It consisted of estimating the average band power of each of the two bipolar channels in a predefined, subject-specific reactive frequency bands by (1) digitally band-pass filtering the EEG using a 5th- order Butterworth filter, (2) squaring each sample, and (3) averaging over several consecutive past samples. A total of 100 samples wee averaged, getting an estimation of the band power for an interval of 500 ms.

The reactive frequency band of each participant was manually identified after comparing several bands. For each channel and each trial, we computed the difference between the power spectra of two 1-s intervals: an active interval, during which one of the two mental tasks had taken place, with respect to a 1-s baseline interval between 0.5 and 1.5 seconds after the beginning of the trial [8]. Then, we bandpass filtered that difference signal and used the resultant average power as a feature. The features from both bipolar channels for all trials were used to compute the accuracy time course of a linear discriminant analysis (LDA) classifier by means of a ten-times ten-fold cross-validation scheme. In this way we obtained the estimated maximum accuracy of the classifier for power features from the given frequency band. This process was repeated for several possible combinations of bands. The band that led to the highest estimated accuracy was regarded as the subject's reactive frequency band, and the obtained parameters were used to provide feedback and control the navigation paradigm in the later sessions (B and C).

2.5 Navigation Paradigm

The navigation paradigm used in this study was the same as in [8] (Fig. 1). It included two types of interface: one for navigation and another one to allow for a non-control (NC) state. This latter interface was presented only at the beginning of each navigation phase and was aimed at letting the participant get ready for the navigation task and tackle it only when ready. The NC interface consisted of a semi-transparent vertical blue bar placed in the center of the screen (see Fig. 1a). The bar length was computed every 62.5 ms as a result of the LDA classification. If the classifier determined that the participant was carrying out right-hand MI, the bar extended; otherwise, the bar remained at its minimum size. In order to change from the NC to an intentional control (IC) state, the participant had to extend the bar over a certain selection threshold by performing right-hand MI and maintain it extended longer than a subject

dependant selection time, which was around 1 s. Further details can be looked up in the aforementioned work.

The IC interface comprised a semi-transparent circle divided into three parts or quadrants (see Fig. 1b), which corresponded to the three possible navigation commands, namely, move forward (upper quadrant), turn right (lower right quadrant) and turn left (lower left quadrant). A blue bar was placed in the center of the circle and rotated continuously clockwise, always starting at the beginning of the first quadrant and making a complete round in 15 s. The participant extended the bar in the same way as the NC interface, i.e., by carrying out the right-hand MI task. In order to select a navigation command, he/she extended the bar while it was inside the corresponding quadrant. The command was selected if the bar had remained extended over the selection threshold longer than the selection time. In order to help the participant make the selection, the bar stopped rotating when its extension was above the selection threshold. When interacting through the IC interface in phases B1 and B2, the participants not only did see the bar moving from quadrant to quadrant (i.e., from navigation command from navigation command), but they also received audio cues while interacting with the system. Conversely, in phases C1 and C2 they only received the audio cues. These consisted of hearing a female voice saying 'espera' (i.e., the Spanish word for the verb wait) when the IC interface turned into the NC interface. In the opposite case the voice said 'avanza', 'derecha', and 'izquierda' (i.e., the Spanish words for move forward, turn right, and turn left, respectively) when the rotating bar first stepped into the corresponding quadrant.

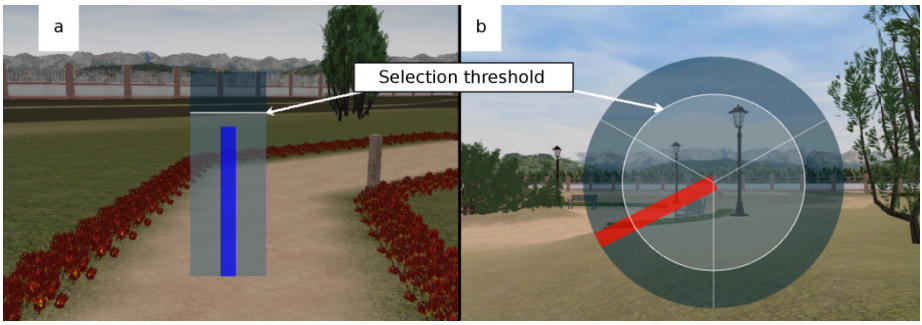


Fig. 1. NC interface (a) and IC interface (b)

2.6 Virtual Environment

As mentioned before, we developed a VE simulating a quiet park. This VE was used in phases B1 and C1. We also developed several distracting stimuli that were added to the park in phases B2 and C2 in order to increase its realism and thus see whether they hampered user performance.

As we aimed at comparing the performance in both VEs, we forced the same path to be followed by the users. 23 commands (the first and the last one are considered neutral, so they are not consider for the evaluation) including advances and turns completed this path. Each individual task comprised three steps: i) users receive a cue

indicating what command should be selected (a red arrow down in the ground); ii) 1s later, the IC interface (visual and auditory or only auditory, depending on the phase) appears letting the subject choose a command; iii) after the selection of a command, a delayed feedback is provided (visual and auditory), indicating the users if they performed well or not the last individual task (Fig. 2).

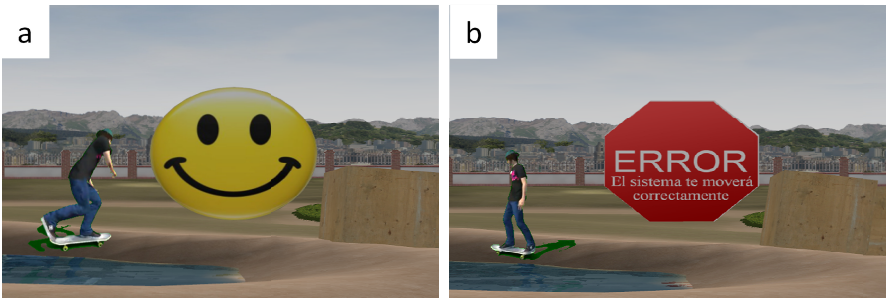


Fig. 2. Positive (a) and negative (b) feedback, accompanied with a ringing bell and a electronic buzz, respectively

The duration of each task depended on the subjects, as it is the time necessary for them to select a command. If users did not select any command after two complete turns of the bar in the control interface, the system considered that the user wanted to switch back to the NC interface (what was considered a mistake in the navigation task). After the selection of a command, the virtual wheelchair moved in a pre-determined way: whether the user chose the right command or not, the demanded movement was performed (the negative feedback alerted them of that decision with some text on it), so all subjects went through the same path (several stimuli happened in a specific part of the park).

Visual and auditory distracting stimuli. We used two kinds of distracting stimuli in phases B2 and C2. There were general stimuli, which happened randomly and could be present in every trial, namely, a plane flying over over the park, a car driving around it and bird tweeting. In all cases the participant also heard their respective particular sounds coming and going. There were also seven specific distractors that happened just as the participant started a trial and before the IC interface was shown:

1. "Skater and boxes falling": a young boy skating appears behind some boxes, making them fall. As the VE supports physics events, the way they fall was different each time. The sound of the falling boxes and the skate remains for a few seconds (see Fig. 3-1).
2. "Rain": The sky gets cloudy, users see (stereoscopically) how the rain falls and makes some puddles on the ground as well as they hear the sound of raining and storming (see Fig. 3-2).
3. "Wind": The trees and the bushes at the park move irregularly and a windy noise is generated. Unfortunately, this dynamic visual effect cannot be appreciated in Fig. 3.

4. "Person reading": The participant approaches a guy who is reading a magazine. While standing in front of him/her, the guy makes some sounds while he glances through the pages and adopts different attitudes like getting interested or bothered by the participant's presence (see Fig. 3-4).
5. "Little boy playing": A child appears from behind some bushes running in zig-zag after three soccer balls that move semi-randomly, depending on the VE's physics (see Fig. 3-5). The cries of several children are also heard.
6. "Dog": A small dog keeps crossing and barking in front of the participant (see Fig. 3-6).
7. "Snow falling": The whole park whitens as the snow falls (see Fig. 3-7).

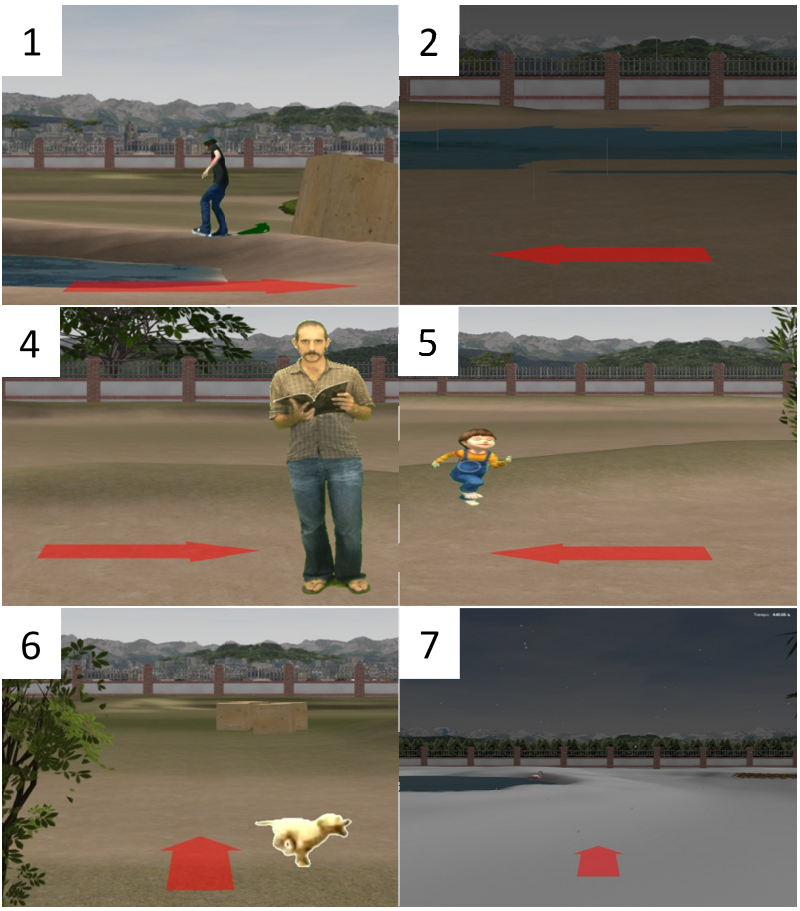


Fig. 3. Distracting stimuli ("Wind" not shown). These images correspond to phase C2, in which the visual interface had been removed

Next, Fig. 4 and Table 2 show the kind of stimulus that was presented at each point and the command that participants had to select. The white lines show the path that users followed and the two black lines represents the two neutral commands (starting and finishing forward movements).

Fig. 4. Path and reference points

Table 2. Navigation task's details. The correct navigation commands and the point in the path is shown for each trial, together with the number of the distracting stimulus that was presented in that trial in phases B2 and C2. See text for details.

Trial order	Point	Command	Distracting stimulus	Trial order	Point	Command	Distracting stimulus
1	a	Right	1	12	d	Forward	6
2	a	Forward	2	13	e	Left	1
3	b	Right	2	14	e	Forward	5
4	b	Left	2	15	f	Left	6
5	b	Left	3	16	f	Right	5
6	b	Forward	3	17	f	Forward	1
7	c	Right	3	18	g	Right	6
8	c	Forward	4	19	g	Left	7
9	D	Right	4	20	g	Forward	7
10	d	Left	5	21	h	Right	7
11	d	Left	4				

3 Results and Discussion

Table 3 shows the reactive frequency bands and the estimated minimum classification error for each participant, along with the sequence of experimental phases they carried out. A two-tailed Mann-Whitney test indicated that there was not a significant difference regarding the estimated minimum classification error between participants that undertook the first sequence (A-B1-B2-C2-C1) (*Mdn* = 20.25%) and those that undertook the second sequence (A-B2-B1-C1-C2) (*Mdn* = 22.5%), $U = 13$, $p = .792$, $r = 0.048$.

Table 3. Phase order, reactive frequency bands (RFB) and minimum estimated error rate (E) after calibration

Participant	RFB (Hz)	E (%)
Phase order: A-B1-B2-C2-C1		
1	9–13	17.7
2	12–14	27.3
3	10–13	24.3
4	10–14	22.6
5	10–14	17.9
6	10–14	13.3
		M = 20.5, *SD* =5.13
Phase order: A-B2-B1-C1-C2		
7	11–14	28.8
8	10–12	22.5
9	10–14	9.0
10	10–12	24.5
11	10–13	20.7
		M = 21.1, *SD* = 7.41

The average percentage (standard deviation in parentheses) of correct selections from a total of 21 trials for the sample of 11 participants was 66.2% (22.1%) for B1, 67.1% (22.3%) for B2, 66.7% (21.4%) for C1, and 67.9% (15.1%) for C2. A two-way repeated measures ANOVA with interface (visual and auditory, auditory) and presence of distractors (true, false) as within-subjects factors revealed that there was not a significant main effect of interface, $F(1, 10) = 0.009$, $p = .924$, $\eta p2 = .001$, nor a main effect of the presence of distractors, $F(1, 10) = 0.096$, $p = .763$, $\eta p2 = .010$. The interaction between these factors was not significant as well, $F(1, 10) = 0.007$, $p = .933$, $\eta p2 = .001$.

These results suggest that the extent to which the participants controlled the navigation application did not depend on the type of interface they used (visual and auditory against purely auditory) nor on whether or not there were distracting stimuli present during the task. Therefore, it might be concluded that the degree of control achieved in a controlled environment—the type of environment most frequently used for training—suggest and encourage the experimental driving of a real wheelchair in a daily environment.

Nevertheless, the relatively small sample size of our study may limit the validity of this conclusion. It would be convenient to extend the number of participants. Besides,

the distracting stimuli we used in our study could obviously not cover the whole range of events that may interfere with controlling the BCI. We particularly did not use stressful or emotional stimuli, which might have a stronger impact on user performance.

In [12], authors verified the speech effect and cognitive distractions while users controlled a robotic prosthesis. They concluded that if one user had reached a good control over the system, he/she could carry out different cognitive tasks without worsening their BCI control. With a similar objective, Friedrich and colleagues [13] tested a four-class BCI with both active and passive auditory distractions. Results showed that these distractors did not affect user performance.

On the other hand, it is known that user motivation positively affects his/her performance when using BCI systems [14]. Therefore, driving a wheelchair in a real environment rather than a controlled one could compensate for the negative influence of distracting stimuli. In this regard, it would be very interesting to carry out similar tests with real users of this navigation application, such as patients of amyotrophic lateral sclerosis (ALS) or stroke.

Acknowledgements. This work was partially supported by the Spanish Ministry of Science and Innovation, project TEC 2011-26395, the European fund ERDF and by the University of Málaga. We would like to thank the volunteers for their time and contribution to the experiment.

References

1. Neuper, C., Scherer, R., Wriessnegger, S., Pfurtscheller, G.: Motor imagery and action observation: Modulation of sensorimotor brain rhythms during mental control of a brain-computer interface. Clinical Neurophysiology **120**(2), 239–247 (2009). doi:10.1016/j.clinph.2008.11.015
2. McCane, L.M., Sellers, E.W., Mcfarland, D.J., Mak, J.N., Carmack, C.S., Zeitlin, D., et al.: Brain-computer interface (BCI) evaluation in people with amyotrophic lateral sclerosis. Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration **15**(3–4), 207–215 (2014). doi:10.3109/21678421.2013.865750
3. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology **113**(6), 767–791 (2002)
4. Leamy, D.J., Kocijan, J., Domijan, K., Duffin, J., Roche, R.A., Commins, S.R., et al.: An exploration of EEG features during recovery following stroke - implications for BCI-mediated neurorehabilitation therapy. Journal of NeuroEngineering and Rehabilitation **11**(1) (2014). doi:10.1186/1743-0003-11-9
5. Lotte, F., Larrue, F., Mühl, C.: Flaws in current human training protocols for spontaneous brain-computer interfaces: Lessons learned from instructional design. Frontiers in Human Neuroscience, (SEP) (2013). doi:10.3389/fnhum.2013.00568
6. Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., Pfurtscheller, G.: Brain-computer communication: Motivation, aim, and impact of exploring a virtual apartment. IEEE Transactions on Neural Systems and Rehabilitation Engineering **15**(4), 473–482 (2007)
7. Ron-Angevin, R., Díaz-Estrella, A.: Brain-computer interface: Changes in performance using virtual reality techniques. Neuroscience Letters **449**(2), 123–127 (2009)

8. Velasco-Álvarez, F., Ron-Angevin, R., da Silva-Sauer, L., Sancha-Ros, S.: Audio-cued motor imagery-based brain–computer interface: Navigation through virtual and real environments. Neurocomputing **121**, 89–98 (2013). doi:10.1016/j.neucom.2012.11.038
9. Kübler, A., Neumann, N., Kaiser, J., Kotchoubey, B., Hinterberger, T., Birbaumer, N.P.: Brain-computer communication: Self-regulation of slow cortical potentials for verbal communication. Archives of Physical Medicine and Rehabilitation **82**(11), 1533–1539 (2001)
10. Guger, C., Schlögl, A., Neuper, C., Walterspacher, D., Strain, T., Pfurtscheller, G.: Rapid prototyping of an EEG-based brain-computer interface (BCI). IEEE Transactions on Neural Systems and Rehabilitation Engineering **9**(1), 49–58 (2001)
11. Guger, C., Edlinger, G., Harkam, W., Niedermayer, I., Pfurtscheller, G.: How many people are able to operate an EEG-based brain-computer interface (BCI)? IEEE Transactions on Neural Systems and Rehabilitation Engineering **11**(2), 145–147 (2003)
12. Foldes, S.T., Taylor, D.M.: Speaking and cognitive distractions during EEG-based brain control of a virtual neuroprosthesis-arm. Journal of NeuroEngineering and Rehabilitation **10**(1) (2013). doi:10.1186/1743-0003-10-116
13. Friedrich, E.V.C., Scherer, R., Sonnleitner, K., Neuper, C.: Impact of auditory distraction on user performance in a brain-computer interface driven by different mental tasks. Clinical Neurophysiology **122**(10), 2003–2009 (2011). doi:10.1016/j.clinph.2011.03.019
14. Kleih, S.C., Nijboer, F., Halder, S., Kübler, A.: Motivation modulates the P300 amplitude during brain-computer interface use. Clinical Neurophysiology **121**(7), 1023–1031 (2010)

Real-Time Monitoring of Biomedical Signals to Improve Road Safety

José Miguel Morales[1](✉), Leandro Luigi Di Stasi[2], Carolina Díaz-Piedra[2], Christian Morillas[1], and Samuel Romero[1]

[1] Brain Computer Interface Lab, Department of Computer Architecture and Technology, University of Granada, Granada, Spain
jm3661@correo.ugr.es, {cmg,sromero}@ugr.es
[2] Mind, Brain, and Behavior Research Center (CIMCYC), University of Granada, Granada, Spain
{distasi,dipie}@ugr.es

Abstract. Fatigue at the wheel has been strongly related to car accidents. Traditionally, this phenomenon has been studied in laboratory conditions by means of performance testing. Here, we aimed to improve road safety assessing driver fatigue at the wheel in real scenarios. For this purpose, we have built BioTracker®: a flexible non invasive platform. A smartphone and a microcontroller unit are the core of sysetm. In this paper, we describe BioTracker®, and we present some examples of its implementation.

Keywords: Brain waves · Electrocardiogram (ECG) · Electroencephalogram (EEG) · Fatigue · Gaze tracking · Road safety · Wearable

1 Introduction

Fatigue at the wheel can impair driving performance as much as alcohol. Unfortunately, due to inconsistent definitions and the lack of reliable testing devices (cf. breathalyzer), it is extremely difficult to incorporate this concept into either driver support technology or traffic law [14].

A recent study from the American Automobile Association (AAA) Foundation for Traffic Safety [2] found that 28% of drivers admitted that they used to drive tired. In Europe, a recent survey of car drivers showed that these statistics vary between countries: from 6.1% in Croatia to 34.7% in the Netherlands [12]. Consistently, European data show that fatigue-related accidents occur in 10% to 20% cases [4,9] (see Figure 1). Moreover, recent simulations in the USA, estimated that the costs of these accidents have a combined value up to $31 billion [17]. In Spain, nearly 70% of drivers affirmed that they have experienced feeling sleepy while driving [12]. More importantly, 30% of Spanish road accidents could be easily prevented with the placement of effective interventions against fatigue [8].

Road safety experts suspect that even this disturbing picture underestimates the number of accidents or near-miss accidents due to fatigue driving because

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 89–97, 2015.
DOI: 10.1007/978-3-319-19258-1_8

of drivers being unaware of, or not admitting to being fatigued at the time of the accident, or police not acquiring that information at the accident site. Thus, strategies aim at studying the contributing factors that lead to decrease road safety are on the agenda of several organizations, such as the American Academy of Sleep Medicine with the National Healthy Sleep Awareness Project [5].

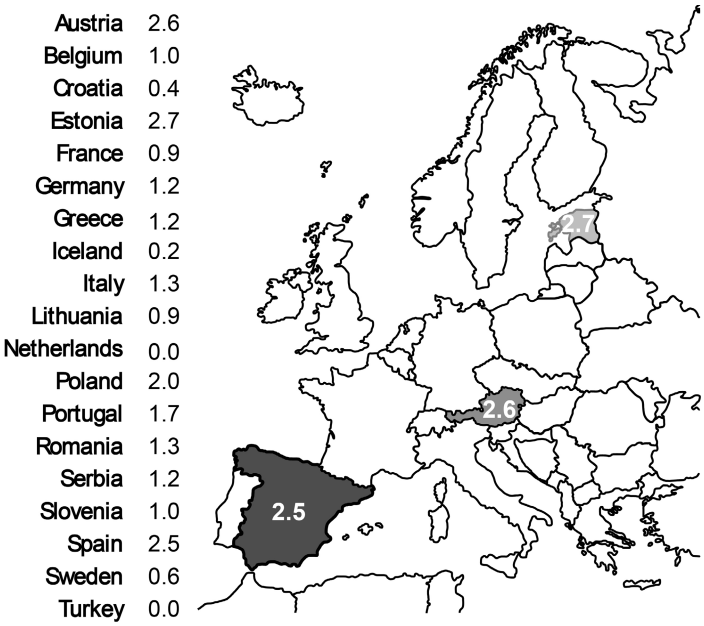


Fig. 1. Proportion (%) of road accidents due to falling asleep at the wheel between 2011 and 2013, each European country [12]

Historically, fatigue research has been dominated by laboratory studies, often using simple tests to measure performance. In contrast, we are planning to assess driver fatigue, using electrophysiological signals (gaze, electroencephalography [EEG] and, electrocardiography [ECG] activities) as fatigue indices while drivers perform demanding real-world tasks.

2 Wearable Recording System

The key principle in the design of the recording system was to minimally disrupt driving tasks. Having this in mind, to make up the platform, we selected and combined commercial off-the-shelf components. The resulted platform is described throughout next sections

2.1 Hardware Elements

The hardware sections of the platform were built around two main computing elements: a Smarphone/Tablet and a microcontroller (Figure 2). The smartphone is a *THL W200* (quad-core 1,5 GHz, 1GB RAM) [10] and the microcontroller is a *Arduino UNO board* (Atmel ATMega328, 16 MHz, 14 digital I/O pins and 6 analog input pins)[1].

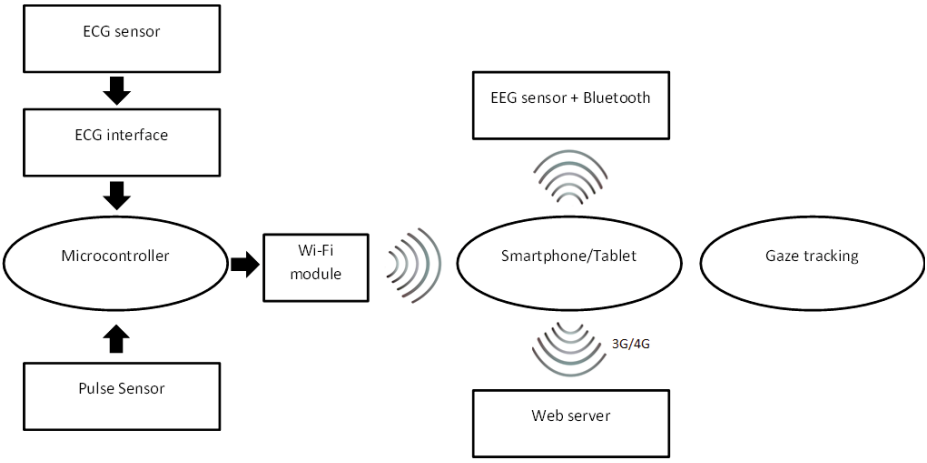


Fig. 2. Main hardware blocks of the recording system. The smartphone *THL W200* and the microcontroller *Arduino UNO board* collects the data form the differents sensor (EEG, ECG, and pulse sensor) and sends these to the web server through 3G/4G.

The microcontroller unit (MCU) collects data from two sensors. Heart-rate is detected using a *PulseSensor* [7] device, directly connected to one of the *Arduino* analog inputs. ECG is recorded by a set of three electrodes connected to an *eHealth Shield* [3] for *Arduino*. An *RN-XV* (Figure 3) Wi-Fi Shield is superposed on the previous shield allow *Arduino* to send information to the smartphone/tablet.

EEG recording is collected throught the *NeuroSky MindWave Mobile* [6] (Figure 4A, 4B) headset which uses with a single frontal dry electrode to record brain waves which includes a Bluetooth link employed, in our case, to send EEG data to the smartphone/tablet.

Gaze behavior is collected using a *Tobii Glasses 2.0* [11].

As the smartphone/tablet only allows for one active connection, in order to connect both the EEG sensor and to the MCU, we chose a combined Bluetooth/ Wi-Fi scheme. The connection from the smartphone/tablet to the web server uses a 3G/4G given that the Wi-Fi connection is busy collecting information from the MCU. Once the connection type it is programmed, the *Arduino* script selects the Wi-Fi network to which it connected. The connection of the *PulseSensor*

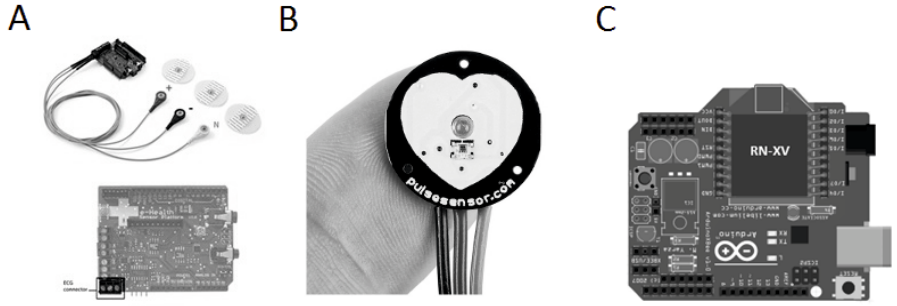


Fig. 3. Elements connected to the microcontroller unit: (A) electrocardiogram Shield, (B) *PulseSensor* (http://pulsesensor.com/), (C) Wi-Fi Shield. The microcontroller unit records the data of the two sensors (A, B) and sends these data through the Wi-Fi shield to the smartphone/tablet.

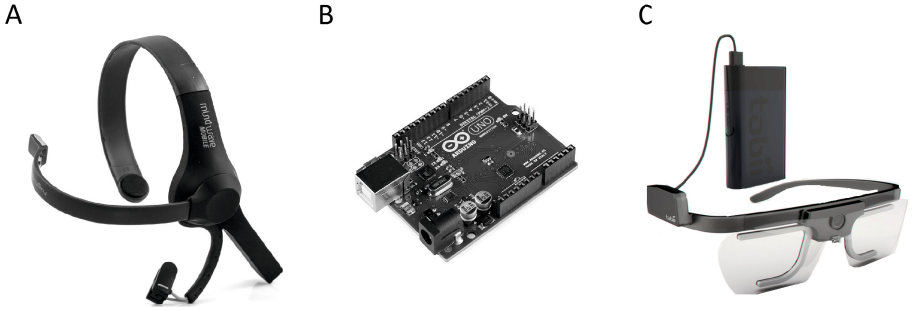


Fig. 4. (A) *Neurosky Mindwave Mobile* used to record electroencephalographic [EEG] activity (http://store.neurosky.com) (B) *Arduino UNO* microcontroller unit (adapted from http://en.wikipedia.org/wiki/Arduino), (C) Tobbi Glasses 2.0 (adapted from http://www.mynewsdesk.com)

consists of three terminals, it will be, GND (ground), V+ (power supply) and ANALOG (analog input). The sampling rate for *PulseSensor* is two milliseconds signaled by an interrupt, measuring systole, and diastole.

2.2 Software Elements

The smartphone/tablet collects the EEG and ECG data, meanwhile, the Tobii Glasses 2.0 records the gaze behavior. physiological and eye movement data are saved in a local database and in the Tobii Glasses 2.0 recording unit, respectively. This solution allows to perform remote offline analysis. Furthermore, this information could be sent to a web server to allow remote querying. The basic software architecture is in exhibited in Figure 5.

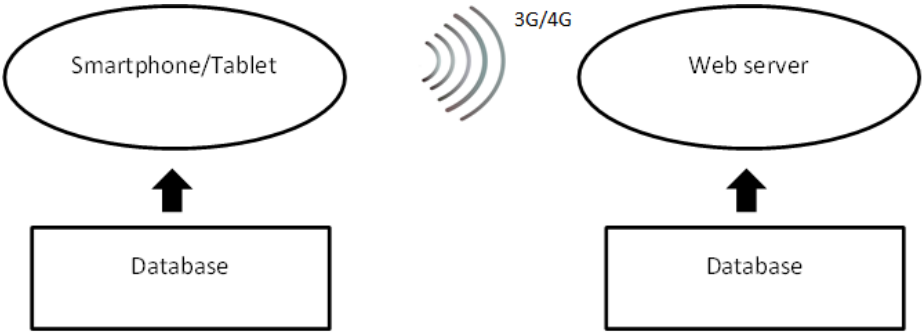


Fig. 5. Main software blocks of the recording system. The smartphone/tablet saves local data about the sensors and user profile, and sends through 3G/4G these information to the web server which saves information locally for later offline analyses.

Smartphone App. The *THL W200* smartphone terminal runs an application under *Android* operating system designed to collect data from the two sensors (EEG and ECG).

The application allows *in-situ* visualization of the recorded biosignals. Aditionally, it can send the information to the remote web server.

The application consits of a main screen (user logging) and three tabs (Figure 6). Tabs visualizes brain activity, Bluetooth connection state, ECG values and a switch to send data to the web server. These biosignals could be shown both in real-time instant values and evolution charts.

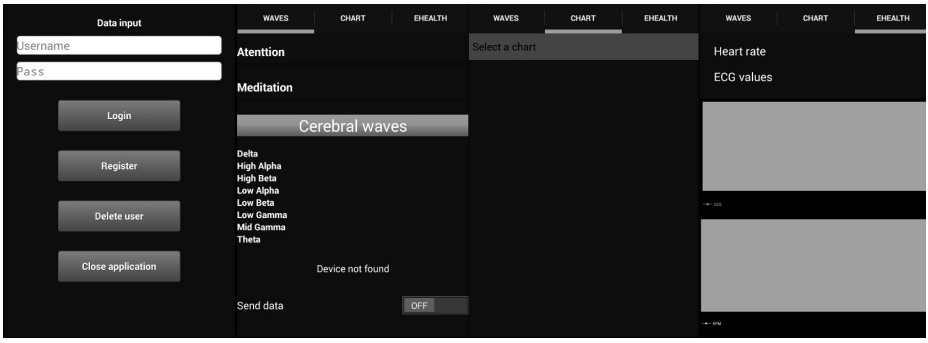


Fig. 6. Smartphone app tabs. From left to right: login tab, where user can register and log in into the app; cerebral waves tab, where brain activity is shown; real-time charts tab, where is possible to see the data of the differents charts in real-time; ECG, heart rate tab. It is possible to see real-time data of ECG sensor.

Two packages manages the application: *adapter*, and *biotracker* containing a single class and eight classes respectively.

The first package is responsible for managing the tabs of the application (sliding between tabs).

The second package is in charge of managing the smartphone/tablet database to record biosignals (brain waves, ECG values and heart-rate in beats per minute). It also includes the managing of users profiles, GPS parameters and real-time charts.

The application uses Bluetooth, GPS, Wi-Fi and 3G/4G data connection.

Web App. The web server receives all drivers profile and biosignals data, and creates a list of routes. Each route (see Figure 7) associates a sequence of GPS coordinates to the biosignals recorded at that location, so a geographical tracking of the evolution of the drivers biosignals could be done.

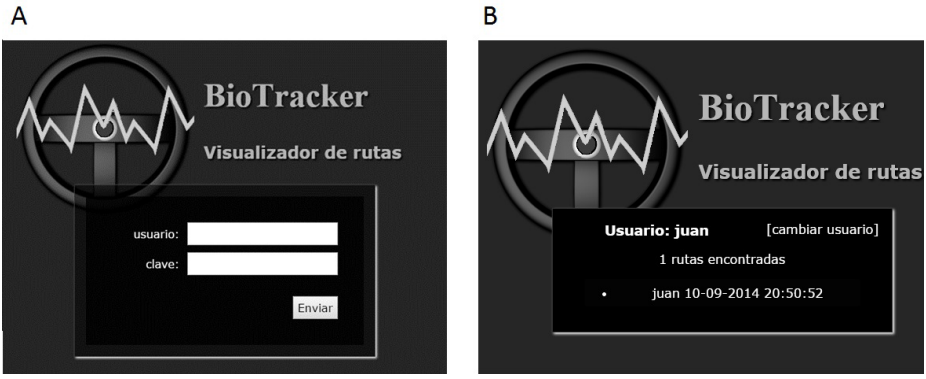


Fig. 7. Web app: (A) user login and (B) routes recorded from a user

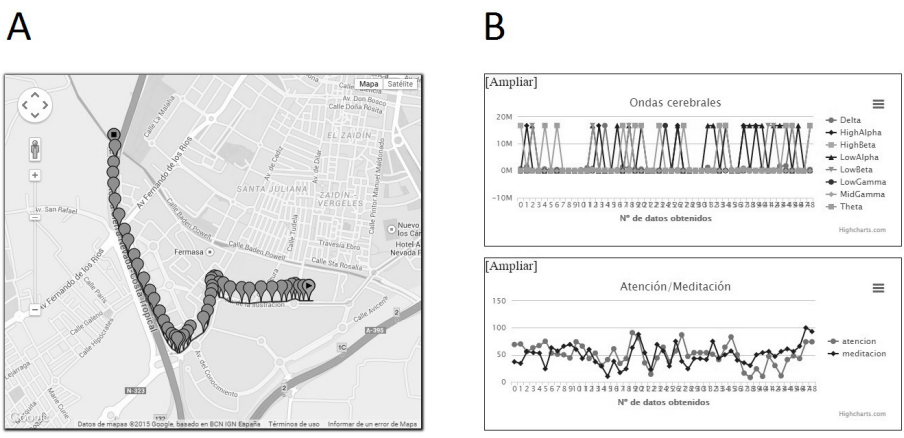


Fig. 8. Example of a route: (A) map showing the route. The coordinates (altitude, latitude and speed) of the route are shown (B) Brain activity records while driving.

The website contains a database that stores the values sent by the Smartphone application. Upon receipt of these data, these are represented in different charts and maps, for later analysis.

When the user logs in, the web shows a list with all routes recorded. Then an individual route can be selected showing the map and the charts associated to this route (see Figure 8). Some additionals parameters as altitude, latitude and speed could be shown.

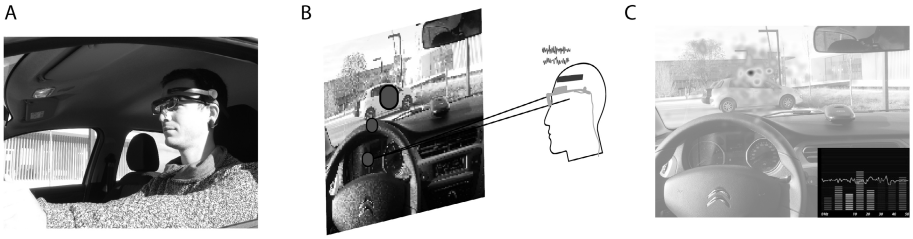


Fig. 9. (A) Integrated solution worn by a driver: Neurosky MindWave and the Tobii Glasses 2.0. Neurosky MindWave is used to record EEG activity. The Tobii Glasses 2.0 eye tracker system is used to record gaze behaviors (B) The eye tracking device integrated with a forward facing scene-camera able to capture what the pilot sees. Gaze fixations are illustrated by circles, with circle diameter indicating fixation duration. Eye movements are recordings with a sample rate of 30 Hz. One channel of EEG is acquired with a sample rate of 512 Hz. C) Heat map generated from the eye tracking over the driving time. Raw brainwaves and power spectrum [0-50 Hz] (lower right side).

3 Conclusions

Very few studies have investigated the effects of fatigue at the wheel on neuroergonomic indices due to environmental constraints (see Introduction). Furthermore, related previous research has used these physiological measurements separately [13]. However, by using EEG or ECG recordings solely, we are blind to the visual scanning techniques that the driver performs, whereas by using gaze behavior recordings in isolation, we lose the direct assessment of the drivers psychophysical state. Recent studies indicate that neither a decline in brain activity [18] nor visual scanning techniques [15], by themselves, predict accidents. Furthermore, cardiorespiratory indices seem to fail to produce reliable results because they are hyper/hyposensitive to external factors [16]. Thus, the combination all these is a research priority.

In order to allow testing all these neuroergonomic indices in a minimally invasive manner, a specific wearable platform is required. Such a platform has been developed and described in this paper.

No studies to date have combined the sensitivity of the EEG activity recordings with the accuracy of the gaze behaviour recordings and ECG while driving, using wearable devices. We expect to employ this platform (see Figure 9) to

obtain results that will represent a significant advance in road safety. This research holds the potential to determine the usefulness and validity of combined neuroergonomic indices as biophysiological markers of fatigue.

The platform is currently being adapted to perform fatigue tests in drivers, both in simulated scenarios and in real driving in closed circuits. Future versions of the platform will seek minimal invasivity for real driving applications.

Acknowledgements. This work has been carried out with the financial support of:

1. Project P11-TIC-7983, Junta of Andalucia (Spain), co-financed by the European Regional Development Fund (ERDF).
2. Project SPIP2014-1426, Dirección General de Tráfico (DGT), Spanish Ministry of Interior.

References

1. Arduino UNO product website. http://arduino.cc/en/Main/ArduinoBoardUno
2. Drowsy driving fact sheet. https://www.aaafoundation.org/sites/default/files/2013%20DDPW%20Fact%20Sheet_2.pdf
3. eHealth product website. http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical
4. European accident research and safety report. http://www.volvotrucks.com/trucks/lithuanian-market/ltlt/NEWSMEDIA/Publications/Safety/Documents/European%20Accident%20Research%20and%20safety%20report%202013.pdf
5. National healthy sleep awareness project. http://www.sleepeducation.org/docs/default-document-library/drowsydriving-health-advisory.pdf?sfvrsn=0
6. Neurosky products website. http://neurosky.com/products-markets/eeg-biosensors/hardware/
7. PulseSensor website. http://pulsesensor.com/
8. Real Automovil Club España. Campaña para prevenir la fatiga en motocicletas. http://www.fatigayconduccion.com/materiales/folleto_motos_2013.pdf
9. Safetynet. fatigue. http://ec.europa.eu/transport/road_safety/specialist/knowledge/pdf/fatigue.pdf
10. THL W200 product website. http://en.thl.com.cn/product/thl-w200.html
11. Tobii glasses user manual. http://www.tobii.com/Global/Analysis/Downloads/User_Manuals_and_Guides/Tobii%20Glasses%20User%20Manual.pdf
12. Wake-up bus sleep study: A survey of 19 european countries. http://www.esrs.eu/fileadmin/user_upload/ESRS_WakeUp_Bus/Presentations/5._ESRS_Brussels_OCT_15_GONCALVES.pdf
13. Borghini, G., Astolfi, L., Vecchiato, G., Mattia, D., Babiloni, F.: Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. Neuroscience & Biobehavioral Reviews **44**, 58–75 (2014)
14. Di Stasi, L.L., McCamy, M.B., Pannasch, S., Renner, R., Catena, A., Cañas, J.J., Velichkovsky, B.M., Martinez-Conde, S.: Effects of driving time on microsaccadic dynamics. Experimental brain research **233**(2), 599–605 (2015)
15. Robinski, M., Stein, M.: Tracking visual scanning techniques in training simulation for helicopter landing. Journal of Eye Movement Research **6**(2), 1–17 (2013)

16. Roscoe, A.H.: Assessing pilot workload. why measure heart rate, hrv and respiration? Biological psychology **34**(2), 259–287 (1992)
17. Shahly, V., Berglund, P.A., Coulouvrat, C., Fitzgerald, T., Hajak, G., Roth, T., Shillington, A.C., Stephenson, J.J., Walsh, J.K., Kessler, R.C.: The associations of insomnia with costly workplace accidents and errors: results from the america insomnia survey. Archives of general psychiatry **69**(10), 1054–1063 (2012)
18. Ueno, A., Uchikawa, Y.: Relation between human alertness, velocity wave profile of saccade, and performance of visual activities. In: 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEMBS 2004, vol. 1, pp. 933–935. IEEE (2004)

Brain-Computer Interface: Usability Evaluation of Different P300 Speller Configurations: A Preliminary Study

Liliana Garcia[1(✉)], Véronique Lespinet-Najib[1], Sarah Saioud[2], Victor Meistermann[2], Samuel Renaud[2], Jaime Diaz-Pineda[3], Jean Marc André[1], and Ricardo Ron-Angevin[4]

[1] Team CIH - Laboratory IMS CNRS UMR 5218, Bordeaux, France
{liliana.garcia,veronique.lespinet,jean-marc.andre}@ensc.fr
[2] ENSC - Bordeaux INP, Bordeaux, France
{sarah.saioud,victor.meistermann,samuel.renaud}@ensc.fr
[3] CATIE - Information and Electronic Technology Center of Aquitaine, Aquitaine, France
j.diaz@catie.fr
[4] Dpto. Tecnología Electrónica, Universidad de Málaga, Málaga, Spain
rra@dte.uma.es

Abstract. Brain–Computer Interface (BCI) is particularly relevant as a new way to interact with the outside world for disabled people. Based on P300 event-related potentials (ERPs) BCIs have been frequently used for communication purposes, being the first P300-based BCI paradigm developed by Farwell and Donchin for visual speller. P300-BCI speller studies require a significant attentional demand during sustained long times which could represent fatigue and feeling of increasing workload. The evaluation of workload while using P300-BCI speller requires taking into account the cognitive, emotional and physical state of participant during task. This would help to improve usability of the system. The objective of the study is to evaluate, through objective and subjective measures, three different size of speller in order to analyze effectiveness, cognitive load and user comfort. Three healthy subjects took part in the experiment. The preliminary results suggest that speller size can have different effects on user performance and represent important workload for subjects.

Keywords: Brain-computer interface (BCI) · Usability · Speller · P300 · Matrix · Size

1 Introduction

Brain-computer interface (BCI) systems [1, 2] are devices that transform a user's brain activity into commands that are interpreted by a machine. Such systems offer a non-muscular channel for users to interact with their environment. This is particularly useful for people suffering from neurodegenerative disorders, such as amyotrophic lateral sclerosis, as they can eventually present severe motor disabilities, to the point of losing control of the muscles that are responsible of voluntary body movements, including eye movement and eventually breathing. In some cases, BCIs turn out to be

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 98–109, 2015.
DOI: 10.1007/978-3-319-19258-1_9

the only way for patients to gain some degrees of communication and autonomy in their daily lives.

The most widely used BCI systems are those based on electroencephalographic (EEG) signal recording, due to its non-invasiveness, but also to its good temporal resolution and ease of use. Three types of EEG-based BCI systems have been used for communication purposes, namely those based on: (a) slow cortical potentials (SCPs), (b) P300 event-related potentials (ERP), and (c) sensorimotor rhythms (SMR) [3]. BCIs based on SCP and SMR requires extensively trained users before they could show sufficient control of their brain activity. In contrast, BCIs based on P300 require minimal training as they rely on a common expected human response to infrequent target stimuli, usually visual. The P300 signal, recorded over the central and parietal regions, is a positive deflection of brain wave at a latency of about 300 ms after stimulus presentation.

The main applications of P300-based BCI systems are aimed at communication purposes. They are based on the P300 speller first developed by Farwell and Donchin [4], which is still referenced and intensely studied [5, 6, 7, 8]. In this BCI, a 6 x 6 matrix of letters, arranged in rows and columns, is shown to the subject. The user focuses his/her attention on the matrix element he/she wishes to select as each row and column is flashed (i.e., intensified) randomly, one after the other. After a number of flashes, the symbol that the user has supposedly chosen is presented on screen.

The effectiveness of the P300-based BCI speller system is guaranteed by a number of studies carried out not only on healthy subjects [9, 10] but also on subjects affected by some motor disability [11]. Overall, these studies conclude that the P300 speller processor is an effective communication tool for people who have lost or are losing their ability to write or speak. However, it is still needed to improve the usability of these BCI speller systems. The current ISO definition of usability (9241-11) includes three measures: effectiveness (accuracy and completeness of users achieving set goals), efficiency (the resources expended to complete goals), and satisfaction (the users' attitude) [12, 13, 14]. Frokjaer et al. argue that these components should be considered as separate and independent aspects of usability [15]. The efficiency and satisfaction can be measured through several subjective variables: mental workload, fatigue, motivation, comfort, pleasure to use, etc [16, 17, 18].

Some factors, such as the mental fatigue induced by a long use [19, 20], the sustained attention at a symbol on screen [21], the user´s motivation [6], [19] or his/her frustration due to a mistake can influence the amplitude and latency of the P300 component (See [22] for a review). In this regard, the influence on performance of the temporal and spatial aspects of the user interfaces of these systems is increasingly drawing the attention of researchers [23, 24, 25].

Although much research has been studied parameters that could influence user performance in P300-based BCI applications, some research are still required, particularly regarding to speller size. Some studies have showed how matrix size affects user task performance. Allison et al. [26] carried out a study comparing three different matrix sizes (4x4, 8x8, 16x16). Results indicated that larger matrices evoked larger P300 amplitude, and the matrix size did not significantly affect performance or preference. In this study, size symbol was the same for the three matrices, thus the larger

matrices appeared larger on the screen. On the other hand, a study comparing two different matrix sizes concluded that accuracy was higher for the 3x3 matrix, whereas that P300 amplitude were higher for the 6x6 matrix condition [8].

In spite of these studies related to matrix sizes, there have been no studies in the literature investigating the effect of speller size. In one study, three different screen sizes were investigated: a computer monitor, a global positioning system (GPS) screen, and a cell phone screen [27]. However no information related to size symbol is provided. According to the screen resolution and the different distances from the participants to the screen, visual fields are 6,4º, 3,7º and 3,56º for computer monitor, GPS and cell phone screen respectively, being almost the same for the two smallest screens. Actually, the main objective of this study was to evaluate BCI performance using these three specifically screens, but not to study screen size effect. In order to study the effect of speller size in P300-based BCI performance, different visual fields should be proposed, as they would be very important to justify the different size speller proposals.

The purpose of this study is to evaluate different speller sizes in order to analyze effectiveness (accuracy and completeness of users achieving set goals), efficiency (the resources expended to complete goals), and satisfaction (the users' attitude). We hypothesized that these three factors are different when using small, intermediate or larger speller size.

2 Methods

2.1 Participants

Three healthy subjects (2 Male, 1 female, aged 20.4 years, SD =0.89) took part in the experiment. None all them had previous experience with BCI systems. All subjects had normal vision and gave informed consent through a protocol reviewed by the ENSC-IMS Cognitive team.

2.2 EEG Data Acquisition and Processing

EEG was recorded using gold electrodes placed at positions Fz, Cz, Pz, Oz, P3, P4, PO7 and PO8, according to the 10/20 international system (see Fig. 1). All channels were referenced to the right earlobe, using FPz as ground. The EEG was amplified through a 16 channel biosignal amplifier (g.BSamp, Guger Technologies). The amplifier settings were 0.5 and 100 Hz for the band-pass filter, the notch (50 Hz) was on, and the sensitivity was 500 µV. The EEG was then digitized at a rate of 256 Hz by a 12-bit resolution NI-USB-6210 data acquisition card (National Instruments). All aspects of EEG data collection and processing were controlled by the BCI2000 system.

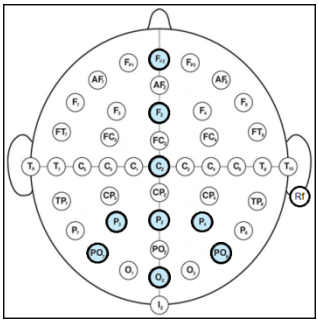


Fig. 1. Electrodes montage for EEG recording

2.3 Experimental Setup

Subjects were seated in a comfortable chair at a distance about 60 cm from the screen. Before the beginning of the experiment, instruction regarding the procedure and BCI speller management was given in verbal form. They were presented with the classical Farwell & Donchin [4] speller, which consists on a 6 x 6 matrix of symbols (36 alphanumeric letters and numbers) arranged within rows and columns (Fig. 2).

Fig. 2. Schematic representation of a classical P300 speller BCI

Each subject participated in a 2 hours session. During this session, subjects were exposed to the three conditions (ie., three different speller size : small; middle and large), whereby 10-15 min break were given between conditions. To avoid effects due to learning, the order in which participants test the three speller sizes was randomized.

Each condition was divided into two phases (see Fig. 3): a first one for calibration purposes and a second one to evaluate the interface. In the calibration phase, the researchers told the participant that he/she would see a number of consecutive random sequences of 20 row and column short intensifications (i.e., flashes). These sequences would be separated by short intervals with no intensifications. To spell a letter or a number, the participant only had to mentally count how many times the desired character had been intensified during a sequence. After these instructions, participants copy-spelled, without receiving any feedback, two French words of 4 characters and a number. These words were “lune”, “feux” “kilo” and the number “2015”. EEG was recorded for offline analysis.

After this runs, we performed a stepwise linear discriminant analysis (SWLDA) of the data from the last three runs to obtain the weights for the on-line P300 classifier.

After calibration and training of the classifier, the evaluation phase started. Participants had to spell 2 words and a number: “CHAT”, “PURE” and “1935”, without spaces. They were instructed not to correct and to continue if the classifier chose a wrong letter. In order to help subject, each word (or number) was showed in the screen for 1 second before to start to spell it. In this phase, the characters typed by the user were displayed inside a text box below the keyboard.

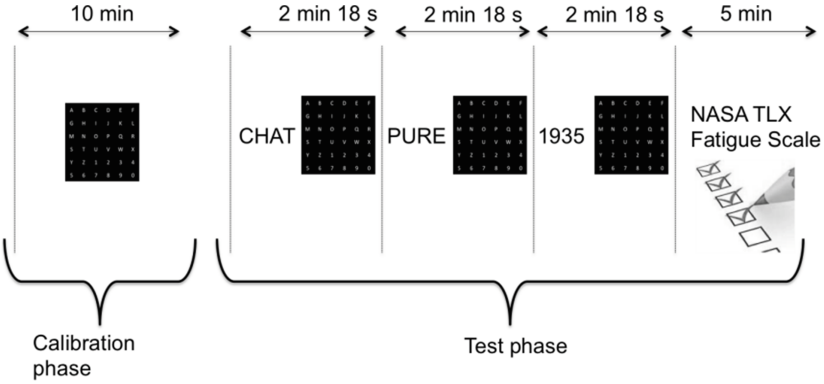


Fig. 3. Experimental design showing the temporal sequence of writing words

Considering that the main objective of the study was to evaluate the performance of three different speller sizes, it was important to take into account the location of the different letters. In this way, if we consider the distance from the letter to the matrix center, 6 different areas are possible (see Fig. 4). The 3 sequences (2 words and 1 number) to spell were chosen such that each area was covered 2 times (by 2 characters per area).

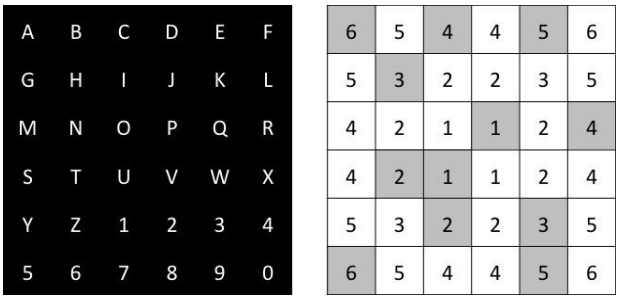


Fig. 4. Area distribution according to the distance from center (left). Letter distribution of the 3 words used: CHAT-PURE-1935 (right)

The values of the temporal parameters for all spellers were based on those used by [9]. Briefly, each row and column was randomly flashed 10 times. Therefore, each character was randomly intensified 20 times. The duration of each flash was 125 ms and the inter-stimulus interval (ISI) between flashes was also 125 ms. There was a pause of 6 s after each sequence of flashes (i.e., after a character had been selected).

This pause gave the subject time to shift attention and gaze to the new character. According to these temporary parameters, the time needed to select a symbol is 36 s – the duration of flashing six rows and six columns each during 125 ms with an ISI of 125 ms, plus the 6s of pause -, being the time to spell the 3 words of the test phase of 7 min and 12 s.

After the evaluation phase of a specific speller, subjects were asked to complete the NASA-TLX test [28, 29] (see Fig. 3). At the end of the session, users were asked to express their preference between the interfaces.

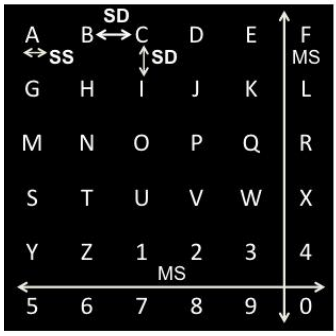
2.4 Speller Size

Three different speller sizes were investigated. All were presented on a 17" TFT screen with a refresh rate of 60 Hz and a resolution of 1440 x 900 px^2. For each speller, the 6 x 6 matrix of 36 characters was centered on the screen.

The biggest size was chosen according to the speller size used in [30], and frequently used by other researchers, i.e. [31]. As in [30], the matrix subtended ±6.98º of the visual field both horizontally and vertically, and symbols were arranged on a grid with a size of 500 x 500 px^2. As proposed in [30], the size of each character was 1.12º (40 px), being the distance (horizontally and vertically) between characters 1.46º (54 px).

The smallest size was chosen according to the smallest symbol size to use without lost of performance. As it is reported in [32], a small symbol size could substantially reduce the spelling performance, being the smallest symbol size recommended 0.4º (25 px). This visual field corresponds to 0.7 cm when situated at 100 cm in front of the screen [32]. Comparing to the symbol size of the previous matrix (1.12º), this small symbol size (0.4º) represents a reduction of 35.89%. In order to maintain the same speller size proportion, the matrix of the smallest speller subtended ±2.51º (185 x 185 px^2), being the distance between characters 0.52º (19 px).

The intermediate size was chosen to be the middle size between the two other spellers. Then, the matrix subtended ±4.75º (350 x 350 px^2), being the intermediate symbol size 0.75º (27 px) and the distance between characters 1º (37 px). The metrics used for each speller can be seen in Fig. 5.



	MS Matrix size	SS Symbol size	SD Symbol distance
Large size	14,69 cm	1,17 cm	1,53 cm
Middle Size	9,98 cm	0,79 cm	1,04 cm
Small size	5,273 cm	0,42 cm	0,55 cm

Fig. 5. Size metrics for each speller size

2.5 Effectiveness Dimension - Objective Measures

To carry out direct comparison across BCIs, performance measures based on classification accuracy of correct target selection with the P300 speller have been considered.

2.6 Efficiency and Satisfaction Dimensions - Subjective Measures

Regarding the degree of efficiency, two variables are evaluated: mental workload and fatigue felt.

- Mental workload: Participants were asked to rate subjective workload using validated NASA Task Load Index (NASA-TLX) [28, 29]. This test represents 6 factors (mental, physical and temporal demands, as performance, effort and frustration) scored between 0-100 in which higher values are indicative of higher workload. Each factor is rated by participant at the end of each matrix speller. The test is composed of 2 parts, in the first part, participants should rate to give a magnitude to each dimensions, then the six subscales were combined into 15 pairs, and for each pair of scales, the subjects were asked to indicate and identify the factor that contributed more to their workload. A weighted average technique was used to compute an overall measure of workload (between 0 and 100) and for analyze the relative contribution of each subscale.
- Fatigue: Simultaneous with NASA TLX test subjects executed the visual analogue scale to evaluate fatigue [33].

Regarding the degree of efficiency, at the end of the session, a final questionnaire based on the comparison of the three speller sizes has been proposed allowing the subject to express their opinions on satisfaction of the interface in terms of fun, ease of use, complexity, etc. A comparative questionnaire adapted from the SUS (System Usability Scale) allowed us to evaluate four dimensions (fatigue, complexity, attractiveness and overall preference). For each dimension, the subject had to rank the three keyboards between them.

3 Results

3.1 Impact of Speller Size on Effectiveness

Performance accuracies for each chosen word and speller size are shown in Table 1. The highest overall accuracy is obtained for large speller size (94.4%). High accuracy variability between words is observed. Effectively, similar accuracy performance is obtained for the first word (CHAT) for all speller sizes (91.6%). On the contrary, different performances are obtained for the third word (1935) for each speller size, being very low when intermediate speller size is used (41.6%).

Table 1. Accuracy performance (%) for each chosen word and speller size

Words	Large Size	Middle Size	Small Size
First: "CHAT"	91.6 %	91.6 %	91.6 %
Second: "PURE"	91.6 %	100.0 %	75.0 %
Third: "1935"	100,0 %	41,6 %	83,3 %
Overall	94,4 %	77,7 %	83,3 %

In order to better understand these results, the total number of wrong selections for each location area and speller size is shown in Fig. 6. So, a higher number of wrong selections are associated to the intermediate speller size, getting 3 letters, located at the bottom, incorrectly selected by two subjects. Regarding the small size speller, the wrong selected characters are more regular distributed over the areas. Only two wrong selections are obtained with the large speller size.

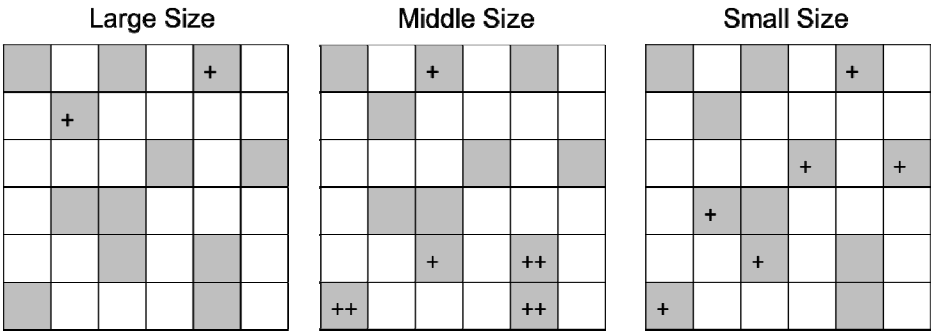


Fig. 6. Total wrong selections according to area location. "+": one error; "++": two errors

3.2 Impact of Speller Size on Efficiency

Average scores obtained for workload (NASA-TLX) and fatigue (visual analogue scale) are shown in Table 2, for each speller size. The results represent an average value, according to type of keyboard and show that the cognitive load is moderately high compared to the task. They seem indicate that speller size did not significantly affect workload scores neither fatigue scores.

Table 2. Average workload (/100) and fatigue (/10) scores obtained

	Large Size	Middle Size	Small Size
Workload Score	53,57	54,33	51,23
Fatigue Score	4,3	4	5

However, as it is shown in Figure 7, three out of six factors considered by NASA-TLX test have an important contribution in workload (score higher than 50): mental, temporal demands and effort.

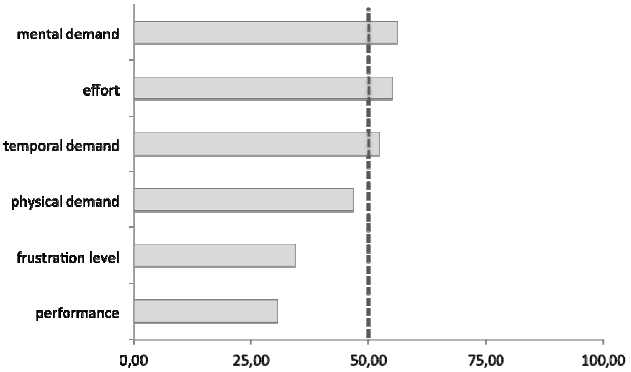


Fig. 7. Average workload scores obtained for each factor of workload

3.3 Impact of Speller size on Satisfaction

Preference for any speller size is very dependent on subjects. The first subject expresses no preference for any speller, the second subject preferred the intermediate size speller and the third one the biggest size. The sample used for this study is too small to advance some conclusion about size speller preference however, the obtained results seem to show individual and specific preference (satisfaction) and so being independent of performance (effectiveness), workload and fatigue (efficiency).

4 Discussion

In this study, three different P300-BCI spellers have been compared through objective and subjective measures. The usability of interface has bean evaluated by users. The three factors of usability were evaluated: effectiveness, efficiency and satisfaction [15].

The results suggest that accuracy seems to be dependent on speller size, getting lower efficiency when using intermediate or small speller size. Although if these are preliminary results, the trends of measures suggest that the large speller size has a better performance, suggesting a higher ability to visual speller task when the subject can realize gaze movement, as it is the case here.

Moreover, a comparison between the numbers of errors between words shows a worsening on efficiency, probably due to user's fatigue or to a lack of attention but only with intermediate and smaller speller size. However, in terms of fatigue experienced (with analogue visual scale), there is no difference between the types of speller size. The level of fatigue experienced by the subject is relatively small.

In the same way, we observed that most of the character errors in each session corresponds rather to the intermediate and smaller speller size, this later presenting a more sparse distribution in the different areas of the speller (as shown in Figure 6). Interestingly, the intermediate and small BCI spellers showed greatest number of errors particularly with the numeric characters.

We observe that visual features dimensions are important in the visual BCI spellers design possibly because using bigger size allows enhanced contrast. These results are in agreement with previous authors, which have showed that large elements can be identified more easily in the periphery [29].

Subjective analysis using the NASA TLX suggest that cognitive factors such as temporal and mental demands, both with effort, could have great contribution on the global workload. Furthermore, although this remains to be demonstrated by additional experiments, frustration level does not seem to have influenced dramatically the execution of the task. This may be explained by the fact that subjects were advised that making mistakes does not mean they were unable to realize task and that the aim of experiment was rather to assess the effectiveness of the interface.

Those overall preliminary results suggest that despite using such interface requires a high degree of concentration; the subjects have a good degree of satisfaction and acceptability.

Our results are coherent with the literature, which establish that the three factors of usability (effectiveness, efficient and satisfaction) are relatively independent [15].

5 Conclusion

We have performed one comparative analysis of performance and usability of different size of P300 BCI speller. The aim of this preliminary work was to identify the best parameters to be considered to enhance adaptability and to develop methods to communicate with the outside world. Although the obtained results are preliminary, these one suggest that subjects' performance during the execution of the task is higher when larger speller size is used. This could be implemented later for people with severe motor disabilities. We noted relative contribution in workload particularly on mental, temporal demand and effort. Further studies will be needed in order to establish the most important advantages for the users.

Acknowledgements. This work was partially supported by the University of Málaga, by the Spanish Ministry of Economy and Competitiveness through the project INCADI (TEC 2011-26395), by the European Regional Development Fund (ERDF). This project benefit from specific funding through The Excellence Initiative of the University of Bordeaux (IdEx Bordeaux). We would like to thank the volunteers for their time and contribution to the experiment.

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology **113**(6), 767–791 (2002)
2. Birbaumer, N.: Breaking the silence: Brain-computer interfaces (BCI) for communication and motor control. Psychophysiology **43**(6), 517–532 (2006)
3. Mak, J., Wolpaw, J.: Clinical applications of brain-computer interfaces: Current state and future prospects. IEEE Reviews in Biomedical Engineering **2**, 187–199 (2009)

4. Farwell, L., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event related brain potentials. Electroencephalography and Clinical Neurophysiology **70**(6), 510–523 (1988)
5. Bianchi, L., Sami, S., Hillebrand, A., Fawcett, I., Quitadamo, L., Seri, S.: Which physiological components are more suitable for visual ERP based brain-computer interface? A preliminary MEG/EEG study. Brain Topography **23**(2), 180–185 (2010)
6. Kleih, S., Nijboer, F., Halder, S., Kübler, A.: Motivation modulates the P300 amplitude during brain- computer interface use. Clinical Neurophysiology **121**(7), 1023–1031 (2010)
7. Krusienski, D., Sellers, E., McFarland, D., Vaughan, T., Wolpaw, J.: Toward enhanced P300 speller performance. Journal of Neuroscience Methods **167**(1), 15–21 (2008)
8. Sellers, E.W., Krusienski, D.J., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R.: A P300 event-related potential brain–computer interface (BCI): The effects of matrix size and inter stimulus interval on performance. Biological Psychology **73**(3), 242–252 (2006)
9. Donchin, E., Spencer, K., Wijesinghe, R.: The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. IEEE Transactions on Rehabilitation Engineering **8**(2), 174–179 (2000)
10. Wang, C., Guan, C., Zhang, H.: P300 brain-computer interface design for communication and control applications. In: 27th Annual International Conference of the Engineering in Medicine and Biology Society IEEE-EMBS 2005, pp. 5400–5403 (2005)
11. Sellers, E.W., Donchin, E.: A P300-based brain–computer interface: Initial tests by ALS patients. Clinical Neurophysiology **117**(3), 538–548 (2006)
12. ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability (1998)
13. Nielsen, J.: What is usability? In: Usability Engineering, pp. 23–48. Academic Press, Cambridge (1993)
14. Nielsen, J.: Heuristic evaluation. In: Nielsen, J., Mark, R.L. (eds.) Usability Inspection Methods. John Wiley & Sons, New York (1994)
15. Frojkaer, E., Hertzum, M., Hornbaek, K.: Measuring usability: are effectiveness, efficiency, and satisfaction really correlated. In: CHI 2000, pp. 345-352. ACM Press, New York (2000)
16. Chanquoy, L., Tricot, A. Sweller J.: La charge cognitive. Edition Armand Colin (2007)
17. Jeng, J.: Usability assessment of academic digital libraries: Effectiveness, efficiency, satisfaction, and learnability. Libri **55**(2–3), 96–121 (2005)
18. Hertzum, M.: Images of usability. International journal of Human-Computer Interaction. **26**, 567–600 (2010)
19. Kececi, H., Degirmenci, Y., Atakay, S.: Habituation and dishabituation of P300. Cognitive and Behavioral Neurology **19**(3), 130–134 (2006)
20. Murata, A., Uetake, A.: Evaluation of mental fatigue in human-computer interaction - analysis using feature parameters extracted from event-related potential. In: 10th IEEE International Workshop on Robot and Human Interactive Communication, pp. 630–635 (2001)
21. Mangun, G.R., Buck, L.A.: Sustained visual spatial attention produces costs and benefits in response time and evoked neural activity. Neuropsychologia **36**(3), 189–200 (1998)
22. Polich, J., Kok, A.: Cognitive and biological determinants of P300: An integrative review. Biological Psychology **41**(2), 103–146 (1995)
23. Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., Wolpaw, J.: Bci 2000: A general-purpose brain-computer interface (BCI) system. IEEE Transactions on Biomedical Engineering **51**(6), 1034–1043 (2004)
24. Lu, J., Speier, W., Hu, X., Pouratian, N.: The effects of stimulus timing features on P300 speller performance. Clinical Neurophysiology **124**(2), 306–314 (2013)

25. McFarland, D.J., Sarnacki, W.A., Townsend, G., Vaughan, T., Wolpaw, J.R.: The P300-based brain-computer interface (BCI): Effects of stimulus rate. Clinical Neurophysiology **122**(4), 731–737 (2011)
26. Shih, J.J., Townsend, G., Krusienski, D.J., Shih, K.D., Shih, R.M., Heggeli, K., Paris, T., Meschia, J.F.: Comparison of checkerboard P300 speller vs. row-column speller in normal elderly and aphasic stroke population. Paper Presented at the Fifth International Brain-Computer Interface Meeting, Asilomar Conference Grounds, Pacific Grove, CA (2013). http://castor.tugraz.at/doku/BCIMeeting2013/020.pdf (retrieved)
27. Allison, B.Z., Pineda, J.A.: ERPs evoked by different matrix sizes: implications for a brain computer interface (BCI) system. IEEE Trans. Neural Syst. Rehabil. Eng. **11**, 110–113 (2003)
28. Li, Y., Nam, C.S., Shadden, B., Johnson, S.: A P300-based brain–computer interface (BCI): effects of interface type and screen size. International Journal of Human-Computer Interaction **27**(1), 52–68 (2011)
29. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. Advances in Psychology **52**, 139–183 (1988)
30. Cegarra, J., Morgado, N.: Étude des propriétés de la version francophone du NASATLX. In: Communication Présentée à La Cinquième édition Du Colloque De Psychologie Ergonomique (Epique) (2009)
31. Treder, M.S.: Blankertz, B,: Covert attentionand visual speller design in an ERP-based brain-commputer interface. Behav. Brain Funct. **6**, 28 (2010)
32. Brunner, P., Joshi, S., Briskin, S., Wolpaw, J.R., Bischof, H., Schalk, G.: Does the "P300" Speller Depend on Eye Gaze? J Neural Eng **7**, 056013 (2010)
33. Salvaris, M., Sepulveda, F.: Visual modifications on the P300 speller BCI paradigm. J Neural Eng **6**(4), 046011 (2009)
34. Kim, E., Lovera, J., Schaben, L., Melara, J., Bourdette, D., Whitman, R.: Novel method for measurement of fatigue in multiple sclerosis: Real-Time Digital Fatigue Score. J Rehabil Res Dev. **47**(5), 477–484 (2010)

Accessing Tele-Services Using a Hybrid BCI Approach

Chris Brennan[1(✉)], Paul McCullagh[1], Gaye Lightbody[1], Leo Galway[1], Diana Feuser[2], José Luis González[3], and Suzanne Martin[4]

[1] Computer Science Research Institute, Ulster University, Coleraine BT37 0QB, UK
brennan-c15@email.ulster.ac.uk, p.j.mccullagh@ulster.ac.uk
http://www.compeng.ulster.ac.uk/csri.php
[2] Institute of Automation, University of Bremen, Otto-Hahn-Allee 1, 28359 Bremen, Germany
diana.feuser@uni-bremen.de
[3] Telefonica de España, Ronda de la Comunicación, s/n, 28050 Madrid, Spain
[4] Institute of Nursing and Health Research, Ulster University, Coleraine BT37 0QB, UK

Abstract. Brain Computer Interface (BCI) technology has achieved limited success outside of laboratory conditions. This technology is hindered by practical considerations of set up, lack of robustness and low Information Transfer Rate (ITR). There are two interfaces in a BCI system: the brain's interface with the computer and the computer-environment interface, which provides access to applications for the user. Three user services were implemented: control of the smart home, entertainment and communication. These may be accessed through a graphical user interface controlled by a BCI. The paper contrasts the performance of an SSVEP based system with a hybrid BCI comprising eye gaze and muscle response (measured at the scalp). The hybrid developed utilizes the EPOC for recording electrical potential and an EyeTribe gaze tracker; these can be combined to provide more robust interaction with applications. Average ITR for the eye tracker and hybrid approaches (190-200 bpm) are higher than for our SSVEP approach (approx. 15 bpm), for the same applications. The poor performance of our SSVEP system was due to the temporal duration of the stimulation (7s) and partly because not all participants could achieve an accuracy of greater than 50%. The current challenge is the replacement of the scalp recorded muscle component with a reliable user modifiable EEG measure.

Keywords: Applications · Brain-Computer Interface (BCI) · Communication · Control · Entertainment · Eye-tracking · hBCI · Hybrid

1 Introduction

The Brain-Computer Interface (BCI) is a paradigm that aims to empower a user's capabilities by providing an input modality to the computer that does

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 110–123, 2015.
DOI: 10.1007/978-3-319-19258-1_10

not require the involvement of the user's peripheral nerves and muscles [1]. In recent years, our increased understanding of human brain function, coupled with advances in technology, has led to significant advances in BCI technology. In particular signal processing algorithms utilize specific patterns in brain activity across both spatial and temporal domains and translate these into classified commands for action by the computer. Increasingly as the algorithms become more complex, the resulting systems may be endowed with more intelligent, contextual processing, thereby leading to augmentation for the user.

The most commonly employed BCI approach, Electroencephalogram (EEG), measures electrical activity recorded along the scalp, resulting from ionic current flows within the neurons of the brain [2]. In this case, the mental state of the user is probed to generate the brain activity patterns that facilitate control, which will vary in accordance with the operating protocol [3]. The most commonly employed approaches include the P300 component of Event-Related Potential (ERP), Event-Related Desynchronisation/Synchronisation (ERD/ERS), motor potentials, Steady State Visually Evoked Potentials (SSVEP) and Slow Cortical Potentials (SCP) [4]. Current limitations include slow Information Transfer Rates (ITR), high inter/intra-subject variability, inconvenient set up procedures, and the need for carefully controlled environments [5]. Such restrictions adversely affect those that could potentially benefit most from this technology, individuals who suffer from disease and traumatic brain injury [6].

To compensate for these limitations and utilize the advantages of different BCI methods, recent emphasis has been placed upon 'hybrid' architectures [7]. One possible architecture may combine BCI approaches (e.g. motor imagery and sensory stimulation) with separate technologies, such as eye tracking [6]. Hybrid BCIs (hBCI) have the potential to increase transfer rates and classification accuracy, thereby promoting the acceptance and adoption of the technology. An hBCI, which combines EEG with eye tracking, may employ inputs sequentially or collaboratively, with each modality responsible for distinct aspects of user input processing or for cooperative command classification [6]. As conventional BCI systems are hindered by an extensive list of limitations, which have prevented exploitation, a hybrid BCI has the potential to overcome these and promote widespread deployment. Such advances could see this technology being utilised in other application areas such as entertainment, neurorehabilitation, neuroprosthetics, immersive education, and other specialist areas. By improving performance and usability through hybrid implementations, advances in other areas can be made, resulting in improvements to the technology overall, which will cycle back to and improve conventional BCI's as an assistive technology for the severely disabled. Current P300 systems, which are used by individuals that have severe mobility reducing conditions such as ALS, can achieve a spelling rate of between 5-10 characters per minute. For some people this is their only means of communication and so, it is of importance to improve upon this technology, thus improving social inclusion, quality of life, and well-being.

This work supports the hypothesis that a hybrid sensor fusion approach will facilitate better human-computer interaction because performance and utility

will be improved. An architecture initially developed for the EU FP7 BRAIN project [8] has been extended to facilitate an hBCI approach. The architecture consisted of a set of tools for interface development and testing, along with SSVEP and ERD/ERS recording. An Intuitive Graphical User Interface (IGUI) tool has been extended to support the combination of eye tracking and pseudo-EEG control through the use of inexpensive, easily deployed devices. It is anticipated that the use of commercially available technology will facilitate better human-computer interaction within the smart environment, permitting access to a number of computer-mediated services. Three trial user services were implemented: entertainment, control of the environment, and communication by iconography.

2 Architecture and Interfaces

In BRAIN the BCI pipeline comprised data acquisition, signal processing (BCI-2000), and user interface control using classified UDP packets. It has since been updated to receive input from multiple input modalities, see Fig 1. The Intuitive User Interface (IUI) is realized as two components; the user interface itself and a programmatic application interface, known as the Universal Application Interface (UAI). The IGUI provides a user interface that interacts with a virtual representation of a domestic environment, providing a BCI channel for domotic control, entertainment, and communication. The IGUI, implemented using the Java programming language, permits operating system independent deployment and provides an interactive menu system that has been specifically structured to provide four-way directional control. Each menu item controls an event that can be triggered within the smart environment.

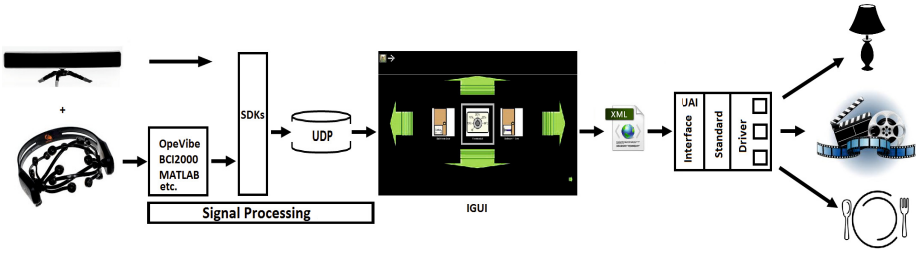


Fig. 1. The pipeline components of the BCI and ET collaborative system for domotic, entertainment and communication control

The UAI interacts with the BCI device through the IGUI for the purpose of providing access to entertainment and communication packages as well as domotic device control. Device interaction may implement a relevant interface such as: X10, Universal Plug and Play (UPnP). UPnP was selected as the communication protocol used in middleware layer to provide interoperable specification with common protocols to other technologies, offering the possibility of

wrapping other technologies. The architecture offers pervasive peer-to-peer network connectivity of PCs, intelligent appliances, and wireless devices. The UPnP architecture is a distributed, open networking architecture that uses TCP/IP and HTTP to control data transfer among networked devices in the home. The design aim was to provide an easily extensible interface that could support various input modalities (BCI paradigms, eye tracking, etc.). The UAI is the second component of the IUI and is responsible for providing a generic platform that forms the bridge between the BCI platform, IGUI, and applications and devices. This interface also facilitates leisure activities such as painting with the users brain waves [9].

3 Control of the IGUI

The first phase of experimentation incorporated a series of preliminary tests, which aimed to evaluate the IGUIs performance in terms of quality assurance. In the first experiment, the IGUI was controlled by features classified from EEG; recorded using the TMSi amplifier, Porti7, and EasyCap combination. Both devices are research or medical-grade products which, when coupled together, provide high quality EEG acquisition. The sampling frequency for the Porti7 amplifier was set to 256Hz and the electrodes were placed at positions PZ, PO3, PO4, OZ, O9, O10 and referenced at site AFZ.

For the BCI paradigm, SSVEP was utilized as the control mechanism and calibration was handled by the BCI-Wizard, which facilitates both frequency and threshold calibration. The calculation of a threshold was computed in BCI2000 for each calibrated or selected frequency. As a quality criterion, an Area Under ROC Curve (AUC) was calculated and if the AUC exceeded a predefined threshold, the calibration was regarded as suitable for further use. The SSVEP signal processing approach takes advantage of the Minimum Energy Combination (MEC) method to estimate power changes in the acquired EEG signals and classify the signal with the highest signal-to-noise ratio (SNR) if the predetermined threshold was exceeded. This algorithm, implemented in C++ as a BCI2000 signal processing module, commissions a spatial filter that readjusts the input channels in order to minimize interference. As a result, electrodes that transmitted poor signals were ignored. Moreover, the combination matrix is constantly adapted to signal changes over time, which is executed on-line in BCI2000. Classification was always based on a two second sliding window and signals were processed every 125 ms.

Stimulation was provided by four Light Emitting Diodes (LEDs) mounted around the outside of a visual interface. Each LED corresponded to a specific direction, left, up, right, down and flickered at 13Hz, 14Hz, 15Hz and 16Hz respectively. The user was required to make a cognitive link between a particular direction and the corresponding LED. For example, to move left the user had to focus on the LED located to the left of the display. This method of control was restrictive, due to the location of the LEDs. When navigating around the menu, the user did not have a full view of the changes they were affecting. In addition to

the physical constraints, some users found this as a cognitive challenge since the directional command arrows and manipulation of menu items caused confusion and difficulties when attempting to navigate to a specific item. Nevertheless, for a control group of five users (N=5), the mean accuracy achieved was 69.6% ±32.9%. However, as a method for measuring the minimum level of tolerated accuracy is yet to be established within the BCI community, we contrasted the achieved results by conducting a BCI accessibility assessment, which provided an indication of the minimum level of accuracy for each user. The individual accuracy values are further expanded in Table 1.

Table 1. Five subjects participated in the BCI accessibility assessment and then completed the command operation assessment. The results, shown in this table, contrast user specific accuracy levels and the levels achieved using a BCI application. Desirable accuracy is the minimum level of control, which a subject deems to be satisfactory and the Acceptable Accuracy is minimum level of control before being considered unusable [10].

User	Desirable Acc. %	Acceptable Acc. %	Achieved Acc. %
1	79	81	16
2	77	81	95
3	77	76	62
4	74	77	80
5	78	77	95

The IGUI was designed to provide four-way directional control, right, left, up and down. A down command takes the user down a level and the up command brings the user back up a level. However, when operating the IGUI at the top level an 'up' command exits the application. When using a BCI paradigm, several unintended exits due to miss-classifications are expected due to this design. The effects of this can be mitigated by using other input modalities or hybrid combinations. In order to provide more intuitive navigation, an eye tracking system was implemented whereby a user could select a menu item by navigating to it with their gaze and dwelling on it as the selection criteria. Essentially, this approach attempts to simulate a more natural input method where the user's gaze acts as the cursor control and the dwell time simulates the mouse click. The Tobii X60 eye tracker was incorporated as the input modality, which offers high performance but is an expensive product (greater than 10K Euro). More recently, however, the EyeTribe Tracker has been incorporated as an alternative input modality because it is available at a much lower cost (less than 200 Euro), and yet retains sufficient accuracy for this application. At first glance the eye tracking-only system appeared to significantly outperform the BCI-only system in terms of accuracy. After extensive testing, however, a number of false positives were detected, especially when a user paused to read or think. This is a recognized limitation of eye tracking technology and reduces the reliability and of course usability of the IGUI.

The next iteration of the IGUI was developed to replace the dwell time component with a Brain-Neuronal Computer Interaction (BNCI) approach. BNCI includes devices that monitor additional physiological signals; not only from signals acquired directly from the brain, but from other measures of nervous activity, such as Electrooculography (EoG), Electromyography (EMG) or heart rate. By implementing the Emotiv EPOC BCI device to acquire EEG activity, it was possible to replace the dwell time with a teeth clench/grind component; as a proxy for an EEG feature. The EPOC was employed to extract the representative features from scalp locations AF3, AF4, F3, F4, F8, F7, FC5, FC6, T7, T8, P7, P8, O1, and O2. The following sections of the paper cover details of the protocol and provide an analysis of the results for each approach.

4 Methods

For each experiment the same stimulation was employed, which consisted of three tasks to be completed by operation of the IGUI. The first represents a domotic task; the second an entertainment task and the third a communication task. The task details are as follows:

Task 1 domotic control: turn on the lamp *(Critical Path = 13)*. For successful completion, each user was required to navigate to the *Dining Room* icon, select it and then navigate to the *Lamp* icon, select it and return to the *Back Garden* icon. The shortest series of sequential operations (i.e. the least commands for successful completion) is: *Right-Right-Right-Right-Down-Right-Right-Down**(Lamp on)**-Up-Left-Left-Left-Left.*

Task 2 entertainment: multimedia control *(Critical Path = 25)*. For successful completion, each user was required to navigate to the *Living Room* icon, select it, navigate to the *Home Media* icon, select it, navigate to the *Home Cinema* icon, select it, navigate to a *video* icon, select it **(play video)**, return to the *Home Media* icon, navigate to the *Controls* icon, select it, navigate to the *Stop* icon, select it **(stop video)** and return to the *Back Garden* icon at the top level. The shortest series of sequential operations is: *Left-Left-Left-Down-Left-Down-Right-Right-Down-Right-Right-Right-Down**(Play)**-Up-Left-Left-Down-Ri- ght-Down **(Stop)**-Up-Up-Up-Right-Right-Right.*

Task 3 communication: indicate hunger *(Critical Path = 7)*. For successful completion, each user was required to navigate to the *Talk* icon, select it, navigate to the *Eat* icon, select it and return to the *Back Garden* icon. The shortest series of sequential operations is: *Left-Down-Left-Left-Down**(Eat)**-Up-Right.*

The sequence of commands enacted for Task 1 (a total of 13 individual commands), for a representative subject, is represented in Table 2. Note that the IGUI must be able to deal with repeated correct commands, due to the variable nature of the subject interaction. It does this by dampening the responsiveness of the interaction. Figure 2 shows a subject enacting Task 1 using the hybrid approach. The IGUI is being operated using the Emotiv EPOC and the EyeTribe Tracker as collaborative input modalities.

Table 2. Sequence of commands enacted for Task 1 for a representative subject. The subject successfully navigated around the interface and turned on the light. They have completed a total of 13 commands but some commands were repeated and disregarded to slow the responsiveness of the application.

Task #1 started			
3 Back Garden 0/0	3 Dining Room 4/4	4 Lamp 7/7	1 Dining Room 9/9
3 Bathroom 1/1	4 Dining Room 4/4	4 Lamp 8/8	1 Bedroom 2 10/10
3 Bathroom 1/1	4 Window 5/5	4 Lamp 8/8	1 Bedroom 2 10/10
3 Bedroom 1 2/2	3 Window 5/5	1 Lamp 8/8	1 Bedroom 1 11/11
3 Bedroom 1 2/2	3 Door 6/6	2 Lamp 8/8	1 Bedroom 1 11/11
3 Bedroom 2 3/3	3 Door 6/6	2 Dining Room 9/9	1 Bathroom 12/12
3 Bedroom 2 3/3	3 Lamp 7/7	1 Dining Room 9/9	1 Bathroom 12/12
3 Dining Room 4/4	11 Lamp 7/7	2 Dining Room 9/9	1 Back Garden 13/13

N.B. Task 1 finished with 13 successful commands and 0 invalid commands as repeated commands were disregarded (e.g. command one was to select the bathroom icon, which was activated twice but only executed once). The subject achieved an accuracy 100% with an average ITR of 16.96 bits/min

5 Results

In BCI it is common to measure performance by calculating ITR in bits/min, which takes into account values for both accuracy and time. According to the literature, traditional BCI systems generally achieved an ITR between 10-25 bits/min [1] whereas conventional input devices such as a mouse and keyboard have been known to achieve an ITR in excess of 300 bits/min [11]. More recent SSVEP-based BCI systems, which use gel electrodes, have managed to achieve an ITR of at least 60 bits/min, [12]. ITR as defined in [13] is shown below:

$$ITR = (log_2 M + P log_2 P + (1-P) log_2[(1-P)/(M-1)]) * (60/T) \quad (1)$$

where M is the number of choices, P is the accuracy of target detection, and T (seconds/selection) is the average time for a selection. This metric can be used to evaluate the performance of the input modalities and compare different experiments. In the BRAIN project motor imagery and SSVEP were used to navigate the IGUI. Kus *et al.* [14] computed an ITR of 12.77 bits/min for a 3-way motor imagery (ERD/ERS) selection task. SSVEP data was then recorded from a control group (N=23), which incorporated the TMSi amplifier and EasyCap for signal acquisition, the SSVEP paradigm as the control mechanism, screen-mounted LEDs for stimulation and the same three tasks, domotic control, entertainment, and communication by iconography. Table 3 presents the results of the data analysis. An important point to note is that when operating the IGUI at the top level an 'up' command exited the application and as such, several unintended exits due to miss-classifications were categorized as inconclusive. In addition, the '-' symbol in the table indicates that a subject attempted to perform the task but could not complete it due to poor performance. Nonetheless, the mean

Fig. 2. A subject enacting Task 1 and operating the IGUI while using the EyeTribe Tracker and Emotiv EPOC as collaborative input modalities. The subject has successfully executed a right command thus scrolling the IGUIs menu from the *Back Garden* icon to the *Bathroom* icon.

accuracy for all participants that completed all three tasks was 79.2% ±14.4% with a mean ITR of 15.23 ±7.92 bits/min.

As mentioned previously, an eye tracking component was implemented to provide better navigational control of the IGUI. The third phase of experimentation focused on a control group (N=10). In this experiment, the SSVEP paradigm was replaced with an eye tracking-only approach. The Tobii X60 Eye Tracker was incorporated as the input modality. When operating the IGUI at the top level an 'up' command still exited the application. As the eye tracker produced a number of false positives, several unintended selections were experienced, which triggered an unintended exit or incorrect command. However, in this case if an unintended exit occurred, the interaction was paused and the experimenter provided a manual override to restart the experiment at the point in which the mistake occurred. If an unintended command occurred, which did not exit the application, the user was required to rectify the error. These values were marked as false positives and were factored in when calculating the accuracy and ITR. Table 4 presents the results using the Tobii X60 eye tracker as the only input modality. Successful completion of all three tasks resulted in a mean accuracy of 78.39% ±10.21% and a mean ITR of 194.15 ±13.34 bits/min.

The fourth phase of experimentation focused on a further control group of 10 participants. In this experiment, the Tobii X60 based control mechanism was substituted for an EyeTribe Tracker based control method. Similar to the previous experiment, the EyeTribe Tracker was the only input modality and the same protocol was employed once more. Again a number of false positives were detected, which triggered an incorrect command that needed to be rectified by the user or in the case of an unintended exit, by the experimenter. If an unintended exit was experienced the interaction time was paused and a manual override provided to return to the point when the unintended exit occurred. All

Table 3. Accuracy and Information Transfer Rate from SSVEP data collected for 23 healthy participants

Subject	Domotic		Entertainment		Communication		All tasks	
	ACC (%)	**ITR (bpm)**	**ACC (%)**	**ITR (bpm)**	**ACC (%)**	**ITR (bpm)**	**ACC (%)**	**ITR (bpm)**
T1	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T2	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T3	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T4	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T5	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T6	73.33	8.95	60.47	6.03	66.67	9.45	66.82	8.14
T7	-		-		-		-	
T8	ERROR		ERROR		ERROR		ERROR	
T9	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T10	-		-		-		-	
T11	68.80	9.89	-		-		INSUFFICIENT	
T12	52.78	5.78	-		66.67	10.38	INSUFFICIENT	
T13	-		-		-		-	
T14	83.33	25.95	90.00	21.15	100	30.38	91.11	20.06
T15	70.59	5.53	-		-		INSUFFICIENT	
T16	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T17	100	36.23	93.33	18.57	88.89	21.69	94.07	25.5
T18	45.16	1.82	65.38	6.56	70.00	8.10	60.18	5.49
T19	-		-		-		-	
T20	90.00	23.70	91.67	17.23	83.33	24.06	88.33	21.66
T21	INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE		INCONCLUSIVE	
T22	86.67	8.37	-		-		INSUFFICIENT	
T23	100	28.44	68.97	6.64	100	18.68	89.66	17.92
Mean:	**77.00**	**15.47**	**78.30**	**12.70**	**82.22**	**17.53**	**79.17**	**15.23**
SD(±):	**18.51**	**11.98**	**14.92**	**7.00**	**14.78**	**8.48**	**14.38**	**7.92**

unintended commands were counted as false positives and commands to rectify a mistake were counted as true positives, since they were intended selections. These values were factored in when calculating accuracy and ITR. Completion of all three tasks for all users equates to a mean accuracy of 93.81% ±4.59% and a mean ITR of 164.97 ±22.88 bits/min.

The final phase of experimentation included the same control group as the previous experiment (N=10), which used the EyeTribe Tracker only. In this phase, the Emotiv EPOC was implemented to function collaboratively with the EyeTribe Tracker thus providing multimodal interaction. The same protocol was used once more and a significantly reduced number of false positives were detected. As a result there were no unintended exits experienced by any participant at any time. However, unintended commands still needed to be rectified by the user. In addition, the hybrid system also experienced a rate of false negatives, which were not present during the eye tracking-only experiments. A false

Table 4. Accuracy and Information Transfer Rate data collected for 10 healthy participants using the Tobii X60 as the only input modality

Subject	Domotic		Entertainment		Communication		All tasks	
	ACC (%)	ITR (bpm)	ACC (%)	ITR (bpm)	ACC (%)	ITR (bpm)	ACC (%)	ITR (bpm)
UU101	36.11	153.43	58.70	161.20	84.62	129.01	59.81	189.58
UU102	81.25	132.32	61.36	185.21	84.62	158.78	75.74	207.57
UU103	100	138.95	90.00	173.04	100	150.13	96.67	198.56
UU104	86.67	151.34	87.10	128.01	91.67	132.95	88.48	166.19
UU105	76.47	175.17	84.38	178.11	84.62	147.44	81.81	212.42
UU106	81.25	147.02	67.50	167.40	78.57	152.38	75.77	198.71
UU107	81.25	160.38	58.70	163.17	73.33	157.52	71.09	199.77
UU108	72.22	126.19	62.79	163.31	78.57	122.75	71.19	181.22
UU109	86.67	124.85	62.79	169.85	100	128.69	83.15	188.91
UU110	92.86	134.61	69.23	168.45	78.57	163.66	80.22	198.53
Mean:	**79.48**	**144.43**	**70.26**	**165.78**	**85.46**	**144.33**	**78.39**	**194.15**
SD(±):	**17.18**	**16.07**	**12.20**	**15.16**	**9.15**	**14.68**	**10.21**	**13.34**

negative is more tolerable as it does not trigger an incorrect command. Nevertheless, such errors are factored in when calculating the accuracy and ITR. For all tasks the mean accuracy was 96.04% ±3.56% and the mean ITR was 207.41 bits/min.

In our tests, the hybrid system achieved the highest performance in terms of both accuracy and ITR. It is difficult to determine which eye tracking system had the better performance since the Tobii X60 system had lower accuracy but a higher ITR than the EyeTribe system. However, there exists a tradeoff between accuracy and ITR, which means that a more accurate system may report a lower ITR since unintended selections need to be rectified by an additional intended selection; thereby increasing the overall bit rate.

6 Discussion

Previous research studies have focused largely on single paradigms in order to facilitate control using BCI systems. These studies have accuracy, performance and reliability limitations [10]. The combination of different input modalities may improve upon these results, thus providing a greater level of control [10]. For instance, an active BCI paradigm, such as SSVEP and/or Motor Imagery (MI), can deliver a method of direct control, whilst a passive BCI paradigm, using error potentials, for example, can potentially provide implicit control [15]. Such a system could benefit from reinforced decisions while autonomously correcting false positive no-control states, thereby improving upon the performance that the use of a single paradigm can offer. There is still a gap between what an acceptable accuracy for a usable system is and what can be readily achieved outside the specialist laboratory [16]. Indeed there is a further difference in the

Table 5. Accuracy and Information Transfer Rate data collected for 10 healthy participants using the EyeTribe Tracker as the only input modality

Subject	Domotic		Entertainment		Communication		All tasks	
	ACC (%)	ITR (bpm)	ACC (%)	ITR (bpm)	ACC (%)	ITR (bpm)	ACC (%)	ITR (bpm)
UU201	100	65.39	96.15	137.11	100	106.27	98.72	127.55
UU202	100	143.43	100	160.74	100	118.77	100	183.89
UU203	93.33	135.50	100	142.37	80.00	131.10	91.11	170.75
UU204	76.47	87.59	93.10	173.12	100	87.79	89.86	153.85
UU205	93.33	114.42	100	132.88	100	74.78	97.78	143.71
UU206	93.33	107.27	100	146.56	88.89	125.17	94.07	159.85
UU207	87.50	164.74	92.59	151.57	88.89	146.04	89.66	191.28
UU208	88.24	163.45	87.10	173.09	88.89	138.35	88.07	203.85
UU209	100	127.04	96.30	136.37	100	96.15	98.77	157.07
UU210	88.24	122.59	93.10	139.75	88.89	101.10	90.08	157.85
Mean:	**92.04**	**123.14**	**95.83**	**149.36**	**93.56**	**112.55**	**93.81**	**164.97**
SD(±):	**7.36**	**31.38**	**4.37**	**14.91**	**7.29**	**23.11**	**4.59**	**22.88**

performance of such systems when comparing users suffering from brain injury and control subjects [15]. The motivation for this research is that solutions based on a hBCI architecture could harness features from complementary technologies [6] thereby closing these gaps.

Both eye tracking systems investigated outperformed pure BCI systems incorporating SSVEP and ERD/S approaches in terms of ITR. However eye tracking as a technology suffers from false positives when the user is in a 'no control' state. Hence there is a need for a hybrid system, where the user can 'initiate' a command in a predictable manner. At present the hBCI uses a teeth clench as a 'proxy' for an EEG controlled switch. This is for pragmatic reasons because the Emotiv suite comes preloaded with classification algorithms for teeth clench. Furthermore, eye tracking technology, which can aid severely disabled individuals by providing them with a level of control, also experiences its own set of limitations. In general, eye tracking uses both infra-red signals and reflected light to determine the trajectory of a user's gaze. This technology is known to provide efficient cursor control but suffers from intended selection restrictions, an area of significant attention in BCI. Earlier research studies have focused on the frequency of blinks and dwell times as the selection criteria, however, the results are not yet promising[17], thus increasing the demand for further innovation.

Subsequently, the collaboration of both BCI and eye tracking as input modalities may mitigate the individual limitations of each and provide a more natural and usable hybrid system. In 2010 Lee *et al.* [18] proposed a method to combine BCI and Eye Tracking for 3D interaction using a bespoke head-mounted eye tracking device with attached EEG sensors. According to the experimental results, the feasibility of the proposed 3D interaction method using eye tracking and BCI was confirmed [18]. The Tobii X60 eye tracker is able to provide

Table 6. Accuracy and Information Transfer Rate data collected for 10 healthy participants using the EyeTribe Tracker and Emotiv EPOC as collaborative input modalities

Subject	**Domotic**		**Entertainment**		**Communication**		**All tasks**	
	ACC (%)	**ITR (bpm)**	**ACC (%)**	**ITR (bpm)**	**ACC (%)**	**ITR (bpm)**	**ACC (%)**	**ITR (bpm)**
UU201	92.86	172.98	100	204.08	77.78	151.20	90.21	225.41
UU202	100	195.73	100	153.66	100	179.52	100	202.86
UU203	100	158.70	96.15	187.24	100	134.64	98.72	208.84
UU204	92.86	183.15	89.29	214.30	100	134.64	94.05	235.99
UU205	86.67	120.09	89.66	132.76	100	122.40	92.11	156.54
UU206	92.86	163.88	100	192.07	100	107.71	97.62	204.83
UU207	88.24	180.60	89.29	157.78	100	122.40	92.51	195.36
UU208	100	163.11	100	207.32	100	141.73	100	223.49
UU209	100	150.56	96.15	179.75	100	128.23	98.72	199.56
UU210	100	167.77	89.29	199.42	100	134.64	96.43	221.26
Mean:	**95.35**	**165.66**	**94.98**	**182.84**	**97.78**	**135.71**	**96.04**	**207.41**
SD(±):	**5.30**	**20.69**	**5.04**	**26.70**	**7.03**	**19.42**	**3.56**	**22.15**

consistent and reliable control, albeit with some false positives. However it is expensive and therefore is only applicable to laboratory based testing. In addition the Application Programming Interface (API) is not open source and as such, this limits its value as a research tool. By combining the Eye Tribe eye tracker with the EPOC (using a teeth clench for selection), the ITR increased to the performance of the much more expensive eye tracker. We aim for an open source system that incorporates inexpensive and commercially available eye tracking and BCI hardware.

The performance improvement, which is evident from the results in table 6 compared with the results in tables 4 and 5, shows that the tested hybrid system outperforms eye tracking only based approaches in terms of accuracy and ITR. It also outperforms the SSVEP only system. When questioned all users reported that the hybrid system provided greater control as it facilitated the ability to pause, think and even talk in between attempting to execute commands. Unlike the eye tracking only approaches the hybrid did not issue false positives, which are intolerable when controlling events in a smart environment. In this case unintentional commands, which execute events such as opening and closing doors, switching lights on and off, flushing the toilet, opening and closing the garden gate, interacting with food preparation devices, and security systems could create a state of bedlam. Seemingly, the burden of montage of an EEG devices is therefore less severe.

7 Conclusion

In this work we have shown the feasibility of using collaborative input modalities based on a hybrid BCI approach. Both of the devices utilized are cost-effective,

highly usable and sufficiently accurate, which makes them particularly suited for deployment outside of laboratory conditions and into 'the wild'. A benefit of the hybrid approach lies in the system ability to interact only when desired by the user. Each user reported full control of the system, i.e. they were able to pause, think, read and talk in between each task without controlling the system, unintentionally. We have also shown that the fusion of BNCI and eye tracking technology reaches the performance of conventional BCI systems and eye tracking-only approaches. Our objective is to investigate how we can replace the BNCI component with a 'true BCI' paradigm, such as SSVEP. It is of importance to maintain high ITRs while maintaining an efficient and robust system that can be widely deployed. Although such components may not meet the performance of research-grade devices, in our tests the Eye Tribe proved sufficiently accurate to provide the desired level of control. By utilizing devices such as the Eye Tribe tracker and the Emotiv EPOC collaboratively, we believe that it is possible to deploy this system outside of the laboratory.

In the next phase of work, low frequency SSVEP stimulation will be incorporated into the IGUI using on screen pattern reversal positioned at each control point. Four different frequencies will be implemented in a collaborative manner with eye tracking. Each selection can only be made if two conditions are true; the gaze based co-ordinates are in an appropriate area and a frequency within a pre-defined threshold is detected in the EEG. Signal acquisition, processing, and classification will be handled within the OpenVibe environment and VRPN will facilitate fusion between the Eye Tribe Tracker and EEG. At first the EEG signal will be acquired using g.Tec's g.USBamp which, is also supported within the current version of OpenVibe. Experimentation will focus on a control group of 20 subjects who will complete the same tasks twice, once using the EPOC and once using the g.Tec device. A comparative study will contrast the two devices and provide an indication of the difference in usability and performance. A further study in conjunction with the Cedar Foundation will test our system on clients with brain injury to establish it's utility as a collaborative assistive technology. A number of challenges may arise: 1) differentiating between frequency bands 2) poor performance by brain injury subjects; and 3) inconsistent frequencies caused by the refresh rate.

References

1. Wolpaw, J.R., Birbaumer, N., Mcfarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain computer interfaces for communication and control. Clinical Neurophysiology **113**(6), 767–791 (2002)
2. Coyle, D., Principe, J., Lotte, F., Nijholt, A.: Guest Editorial: Brain/neuronal - Computer game interfaces and interaction. IEEE Transactions on Computational Intelligence and AI in Games **5**(2), 77–81 (2013)
3. Lotte, F., Faller, J., Guger, C.: Combining BCI with Virtual Reality: Towards New Applications and Improved BCI. In: Proceedings of the 6th International Conference on Foundations of Digital Games, pp. 1–24 (2013)

4. Volosyak, I., Valbuena, D., Malechka, T., Peuscher, J., Gräser, A.: Brain-computer interface using water-based electrodes. Journal of Neural Engineering **7**(6), 066007 (2010)
5. McFarland, D.J., Miner, L.A., Vaughan, T.M., Wolpaw, J.R.: Mu and beta rhythm topographies during motor imagery and actual movements. Brain Topography **12**(3), 177–186 (2000)
6. Mccullagh, P., Galway, L., Lightbody, G.: Investigation into a Mixed Hybrid Using SSVEP and Eye Gaze for Optimising User Interaction within a Virtual Environment. Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion 8009, pp. 530–539 (2013)
7. Allison, B.Z., Brunner, C., Kaiser, V., Müller-Putz, G.R., Neuper, C., Pfurtscheller, G.: Toward a hybrid brain-computer interface based on imagined movement and visual attention. Journal of Neural Engineering **7**(2), 26007 (2010)
8. Thomson, E., Mathews, S., Todd, D., McCullagh, P., Ware, M., Mulvenna, M., Martin, S.: Developing brain computer interfaces with rapid automated interfaces for non experts. Gerontechnology **9**(2), 4017 (2010)
9. Todd, D.a., McCullagh, P.J., Mulvenna, M.D., Lightbody, G.: Investigating the use of brain-computer interaction to facilitate creativity. In: Proceedings of the 3rd Augmented Human International Conference on - AH 2012, pp. 1–8 (2012)
10. Ware, M.P., McCullagh, P.J., McRoberts, A., Lightbody, G., Nugent, C., McAllister, G., Mulvenna, M.D., Thomson, E., Martin, S.: Contrasting levels of accuracy in command interaction sequences for a domestic brain-computer interface using SSVEP. In: 2010 5th Cairo International Biomedical Engineering Conference (CIBEC), pp. 18–21. IEEE (2010)
11. Krepki, R., Blankertz, B., Curio, G., Müller, K.R.: The Berlin Brain-Computer Interface (BBCI) - Towards a new communication channel for online control in gaming applications. Multimedia Tools and Applications **33**(1), 73–90 (2007)
12. Chen, X., Chen, Z., Gao, S., Gao, X.: A high-ITR SSVEP-based BCI speller. Taylor & Francis: Brain Computer Interfaces **1**(3 & 4), 181–191 (2014)
13. Gao, S., Wang, Y., Gao, X., Hong, B.: Visual and auditory brain-computer interfaces. IEEE Transactions on Bio-Medical Engineering **61**(5), 1436–1447 (2014)
14. Kus, R., Valbuena, D., Zygierewicz, J., Malechka, T., Graeser, A., Durka, P.: Asynchronous BCI based on motor imagery with automated calibration and neurofeedback training. IEEE Transactions on Neural Systems and Rehabilitation Engineering **20**, 823–835 (2012)
15. George, L., Lécuyer, A.: An overview of research on passive brain-computer interfaces for implicit human-computer interaction. In: International Conference on Applied Bionics and Biomechanics, ICABB 2010 (2010)
16. Allison, B., Luth, T., Valbuena, D., Teymourian, A., Volosyak, I., Graser, A.: BCI Demographics: How Many (and What Kinds of) People Can Use an SSVEP BCI? IEEE Transactions on Neural Systems and Rehabilitation Engineering **18**(2), 107–116 (2010)
17. Lightbody, G., Ware, M., McCullagh, P., Mulvenna, M., Thomson, E., Martin, S., Todd, D., Medina, V.C., Martinez, S.C.: A user centred approach for developing Brain-Computer Interfaces. In: Proceedings of the 4th International ICST Conference on Pervasive Computing Technologies for Healthcare. IEEE (2010)
18. Lee, E.C., Woo, J.C., Kim, J.H., Whang, M., Park, K.R.: A brain-computer interface method combined with eye tracking for 3D interaction. Journal of Neuroscience Methods **190**, 289–298 (2010)

Authentication of Brain-Computer Interface Users in Network Applications

M.A. Lopez-Gordo[1,2(✉)], Ricardo Ron-Angevin[3], and Francisco Pelayo[4]

[1] Department of Signal Theory, Communications and Networking, University of Granada, c/ Periodista Daniel Saucedo, 18071 Granada, Spain
malg@ugr.es
[2] Nicolo Association, Churriana de la Vega, 18914 Granada, Spain
malg@nicolo.es
[3] Department of Electronics Technology, E.T.S.I. Telecomunicación, Campus Universitario de Teatinos, University of Malaga, 29071 Malaga, Spain
rra@dte.uma.es
[4] Department of Architecture and Technology of Computers, University of Granada, c/ Periodista Daniel Saucedo, 18071 Granada, Spain
fpelayo@ugr.es

Abstract. Cognitive biometrics aims to user authentication (or identification) by direct measure of electrophysiological signals as response to specific stimuli. In the literature, authentication paradigms for network applications are intended for healthy and independent users with complete control of their muscles. This excludes people with severe motor impairment, such as Brain-computer interface (BCI) users. Conversely, BCIs permit communication with users even in extreme impairment conditions, such as those suffering from locked-in syndrome or in advanced stage of Amyotrophic lateral sclerosis. The downside of BCIs is their very poor performance that, measured in terms of throughput and bit error rate, could lead to impracticable authentication. Specifically, current network applications require users to type long usernames and passwords formed with characters chosen from a large dataset. This forces long BCI sessions that users can not afford due to their heavy cognitive workload. In this paper we present some EEG-based authentication approaches and discuss some relevant aspects that a BCI-based authentication approach should consider for users with severe motor impairment.

Keywords: Brain-Computer Interface · Cognitive biometrics · Electroencephalography (EEG) · Brain-area networks · Authentication · Network applications

1 Introduction

Biometrics studies focus on automatic methods for human identification based on intrinsic physiologic features. It is achieved by mathematical and statistical techniques applied to relevant features with the goal of individual identification. The most popular Biometrics method is finger print [1]. However, some authors claim that there is not yet scientific evidence of their uniqueness for each individual [2]. Therefore, other

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 124–132, 2015.
DOI: 10.1007/978-3-319-19258-1_11

alternatives must be found. In this regard, there are proposals such as hand recognition [3], hand geometry [4], iris [5], facial recognition [6], electrocardiogram (ECG) [7].

The cognitive biometrics (CB) approach consists on the analysis of biosignals that correlates with mental or emotional states for user authentication or identification. Different modalities use electroencephalography (EEG), electrocardiogram (ECG) and electrodermal response (EDR). Then, a definition of CB could be the set of methods and technologies for human recognition based on biosignals directly or indirectly generated by thoughts and cognitive processes [8]. Interesting aspects of CB are its robustness against spoofing and the possibility of generation of unique biosignals for each individual. The current challenge in CB is the feature selection improvement and definition of new paradigms for evocation of unique features. Conversely, EEG-based Brain-computer interfaces (BCIs) have also been proposed in CB literature [9][10].

Currently, network applications are part of our lives and permit global interaction and socialization (e.g. social networks, online shopping, cloud storage, etc.). People with severe motor impairment (e.g. locked-in syndrome, Amyotrophic lateral sclerosis, etc.) could use EEG-based BCIs for words spelling [11], Internet web browsing [12], communication and transmission of cognitive information in general [13][14][15][16][17]. Thus, they seem appropriate for identification in network applications too. However, the level of usability and low performance of this technology (e.g. high bit-error rate (BER), low throughput) makes it inappropriate for current authentication paradigms and protocols.

In this paper we introduce some related works in the field of CB and BCI-based authentication and then present relevant aspects of BCI-users that require extra common efforts from the BCI researches and network applications developers for an affordable authentication.

2 EEG-Based Cognitive Biometrics: Related Works

The electroencephalography (EEG) is the recording with non-invasive electrodes located on the human scalp of electrical signals generated by neurons activity. Since field potentials generated by a single neuron are extremely weak, a certain level of synchrony among neighboring neurons is required for a convenient acquisition. The amount of data that can be obtained from an EEG acquisition is enormous. For instance, a montage with 256 electrodes sampled at 1 KHz and 16 bits per sample will generate 153 Mbytes in only five minutes. In addition, the feature space is very large (and somehow redundant) not only for the amount of data, but for the variety of cognitive tasks, paradigms and inter-individual cognitive and physiological differences that definitively condition the final performance.

There is a variety of usable EEG signals and features in CB. No less important is the set of paradigms for a reliable and reproducible elicitation of the EEG features as well as the classification methods. We mention some approaches as follows.

The first step to set up a valid methodology for CB is the assessment of the reproducibility of brain signals and features. In this regards, the authors of [18] extracted three EEG features, namely spectral power, maximum power and frequency in alpha band, from subjects along different sessions and days. The reproducibility of the EEG features was confirmed by obtaining 100% authentication accuracies.

Studies have also focused in the set of suitable features for authentication. One common direction with promising results is the analysis of energy of well-known EEG rhythms, namely delta (<4 Hz), theta (4-7 Hz), alpha (8-12 Hz), mu (8-11 Hz), beta (16-31 Hz) and gamma (<32Hz), as well as spectral bands decomposition. In [9] the energy of alpha band was extracted and classified by means of a LVQ (learning vector quantization) network. In the experiment, a total of four participants were identified with classification accuracy between 72% and 84%. At that time (1999), this was a significant outcome. In 2003, authors of [19] identified people based on analysis of visual evoked potentials (VEP). In this study, VEPs were acquired in a montage with 61 electrodes while participants gazed a picture. EEG signal was denoised by trials averaging (800 times) and gamma band was extracted and classified by using fuzzy logic algorithms and artificial neural networks. The final classification accuracy on 20 subjects was 94.2%. In 2006, the same author proposed an improvement respect to his previous study from 2003 [20]. Background noise was reduced by means of Principal Component Analysis (PCA) and then an ensemble of classifiers, namely fuzzy logic, discriminant analysis and k-nearest neighbor (kNN) was used. The classification accuracy improved to 96.50%. In [21], a combination of features composed by the power spectral density (PSD) in the band 1 to 30 Hz and eleven AR coefficients were extracted for classification of user age and gender. This multi-factor EEG method showed promising results for a multi-level authentication system.

Other EEG features typically used in BCIs are event-related potentials (ERPs). ERPs are positive or negative deflections of EEG that occur as response to an external stimulus. The most popular ERP among the BCI community is the P300 [22]. Reasons for that is its large amplitude, which makes it suitable for robust classification, as well as its easy elicitation. In [23], authentication based on personal identification number (PIN) was proposed. Authentication obtained 100% accuracy with three subjects and was based on P300. A similar approach was discussed in [24].

A different strategy is the use of models to represent EEG activity. In [25] the coefficients of an auto-regressive model (AR) was used to uniquely identify persons. The AR coefficients were analyzed by a discriminant test. Although results showed an extraordinary classification accuracy of 100%, we must mention that this extraordinary result was obtained when reusing the training data set for evaluation. When the whole data set was split between training and evaluation, the classification accuracy decreased to 80%.

In previous paragraphs we have analyzed some relevant EEG features used in user authentication. However, no less important are the psycho-physiological paradigms used to elicit these and other relevant features. Facial detection has been used to elicit user specific cerebral activity based on the assumption that the uniqueness of neural activity in visual cortex is stronger when user is exposed to visual stimulus [26]. In [26] an authentication process was proposed in which, EEG features were extracted while presenting the user a photograph during 5 seconds. Authentication performance was not reported. In [27] three mental tasks, namely relax, mental reading and mental spelling were analyzed under the hypothesis that each user has a unique brain signature for each cognitive task. The identification was performed on 15 participants and the average classification varied from 79.7% to 89.5%. In [28], authors proposed an

authentication paradigm based on five mental imagery tasks, namely relax, mathematical operation, letter composition, imaging of object rotation and number counting. Fractional spectral information was extracted and classified by means of an Exact Radial Basis (RBE) Neural Network. The classification accuracy across mental activity was similar and approximately 80%.

A case of non-paradigm-based authentication was presented in [29]. Authentication was performed based on classification of five features extracted from raw EEG. No stimulus or mental task was needed. The proposal was tested on three situations, i) legitimate authentication; ii) identity spoofing, iii) intruder test (56 subjects ad 31 intruders). The results were equal error rate (EER) of 3.4% (true acceptance rate of 96.6% and a false acceptance rate of 3.4%).

Also, classification methods and algorithms play an important role in authentication. For instance, in one of the studies previously mentioned [28], authors stated that the results provided could be improved to nearly 100% by using more data and applying the voting rules for authentication. Authors in [30] proposed a multiple combination of classifiers and EEG features and fusing the best results to obtain a better performance. The proposal was applied on a set of 50 legitimate users and 20 intruders. The system obtained an Equal Error Rate (EER) (cross-point of False Positive and False Negative curves) of 2.4%.

3 Network Authentication for Disabled People

3.1 General Principles in Network Security and Authentication

Providing security to the communications in a network is traditionally understood as guaranteeing certain security services, among which we can cite: confidentiality, privacy, access control or authorization, authentication, non-repudiation, integrity. Although there is not a complete consensus about which are the essential services, all the authors in the security area agree that authentication plays a fundamental role in security.

The authentication service in network security allows every part of the communication to be sure that the other end of the communication is who it is being claimed to be. Typically, in a client-server network service, not only the client should authenticate, but also the server would need to authenticate to the client.

For the implementation of the authentication service, several strategies have been followed in the last years, ranging from the use of passwords, tokens, passphrases, biometrics, pin numbers, etc. In summary, we could say that authentication relies on "something you know" (e.g., password, pin), "something you have" (e.g. token) or "something you are" (e.g., biometric features). Current trends in authentication involve the use of several of these alternatives (called factors), so that multi-factor authentication appears in the security arena as a fundamental recommendation [31].

In multi-factor authentication, biometric features play an essential role ("something you are"), as reflected in many of current research efforts [32][33][34]. As it is shown below, BCI-based approaches could become a relevant alternative in specific scenarios for authentication.

3.2 Aspect to Consider in a BCI-Based Approach

A BCI-based authentication system is an ideal approach for people with severe motor impairment. Although currently BCIs are an open technology with multitude of applications (e.g. entertainment [35][36], assessment of cognitive workload [37], etc.), originally they were intended for people with intact cognitive and intellectual capacities but without the ability to move their muscles. Then, BCIs provide disabled users a direct communication channel with their brain. Obviously, human beings are not designed to communicate each other by direct extraction of information from brain signals. Then, BCIs are only recommended under extreme conditions of disability.

For an efficient authentication of BCI users in a network application, both sides (BCI and network application developers) must keep in mind some relevant aspects of BCI users. As follows, we cite some considerations when trying to authenticate BCI users in a simple scenario, that is, login by username and password.

Available Bandwidth. In BCI literature, BCI throughput is called Information transfer rate (ITR) and is measured in bits per minute. In BCI literature, studies with very high ITR are in the order of 100 bits/min [38][39]. Let's think of a BCI user (male) trying to login in his email server. He is requested to introduce his username and password with, let's say, 10 characters each one. Both words can include letters (upper and lower case), numbers and characters from a dictionary of, let's say, eighty characters. This means that the total amount of information that the user must introduce, including carriage returns is approximately 140 bits. Therefore, the initial access to the email server would last one minute and a half approximately.

Truly Muscle-Free. Almost one and a half minutes to login in a network application seems too much. This time was calculated assuming a high throughput. In those visual studies participants were allowed to move their eyes gazing at different stimuli sources, thus causing performance boosting. However, users with the ability to gaze do not really need a BCI, but a much faster and simpler technology such as eye-trackers. Truly muscles-free BCIs, have been developed (see discussion about visual vs. auditory BCIs in [40]) at the cost of worse ITR, that in the case of BCI based on slow cortical potentials could be only 4-8 bits/min. (trial length 4-8 seconds, binary classification, 90% accuracy [41][42]). Then, now the initial access to the email server would last more than seventeen minutes. In order to compensate this, the application could accept login words with fewer characters chosen from a reduced character set at the cost of less secure access.

Robustness to Transmission Errors. Typical EEG features are extracted from heavy background noise composed of electrical artefacts and other concurrent brain activity. A typical example is an eye blink. This motor action appears in EEG as a very energetic pulse that masks, during its duration, any relevant feature. This causes drop of performance in terms of classification accuracy that, in turns, causes bit error rate (BER) increasing. A good accuracy in truly muscle-free approaches could be 95%

[43][44]. In our example, the email server could palliate this undesired effect by offering a predictive text (e. g. T9 [45]) capable to correct spelling errors. Obviously the introduction of redundancy entails increasing password vulnerability.

Autonomy and Willfulness. Since brain activity is always present even when we sleep of are unconscious, a way to switch on/off the network authentication process must be provided for BCI users. In the one hand, the BCI must be able to detect the user awareness and willfulness by means of specific features. In the other hand, the email server of our example must provide additional mechanisms to validate legitimate attempts to access the service (e.g. an additional on/off button above the login frame).

Cognitive Workload. A careful design of the BCI paradigm is crucial to avoid user tiredness by cognitive workload. Long trials or trial repetitions improve BER at the cost of ITR and mental fatigue. In this regards, diverse training techniques have been tried [46] and still constitute an open issue. Since inter-trial resting is a key aspect to keep high performance in BCI users, then network applications should disable or modify default timeouts to these users. Then, our email server could also present a checkbox to disable expiration times when spelling the username and password.

4 Conclusions

In this paper we have discussed some aspects to consider when users with severe motor impairment try to gain access to a network application. Authentication standards must be adapted or configured by application developers for BCI-users. Despite their inherent limitations, BCIs are the ideal (perhaps the only) tool for network authentication of users unable to move or control their muscles. Current BCI technology and most advanced approaches of CB are still far from providing a totally secure authentication tool. In this regards, new EEG features, denoising and extraction procedures and classification strategies are demanded from BCI researchers.

Both, application developers and BCI researchers, should make efforts to meet ideas and to cooperate with each other toward the design of applications, services and interfaces fully operational, even for people unable to control their muscles, in the network.

Acknowledgement. This work was supported and co-financed by Nicolo Association for the R+D+i in Neurotechnologies for disability and the research project P11-TIC-7983, funded by the Junta of Andalucía (Spain) and the European Regional Development Fund (ERDF).

References

1. Wayman, J. (ed.): Biometric systems: technology, design, and performance evaluation. Springer, London (2005)
2. Pankanti, S., Prabhakar, S., Jain, A.K.: On the individuality of fingerprints. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), 1010–1025 (2002)
3. Duta, N., Jain, A.K., Mardia, K.V.: Matching of palmprints. Pattern Recognit. Lett. **23**(4), 477–485 (2002)
4. Aghili, B., Sadjedi, H.: Personal authentication using hand geometry. Presented at the International Conference on Computational Intelligence and Software Engineering, CiSE 2009, pp. 1–4 (2009)
5. Daugman, J.: Recognizing persons by their iris patterns. In: Jain, A.K., Bolle, R., Pankanti, S. (eds.) Biometrics, pp. 103–121. Springer US, Boston, MA (1999)
6. Samal, A., Iyengar, P.A.: Automatic recognition and analysis of human faces and facial expressions: a survey. Pattern Recognit. **25**(1), 65–77 (1992)
7. Biel, L., Pettersson, O., Philipson, L., Wide, P.: ECG analysis: a new approach in human identification. IEEE Trans. Instrum. Meas. **50**(3), 808–812 (2001)
8. Revett, K., Deravi, F., Sirlantzis, K.: Biosignals for user authentication - towards cognitive biometrics? Presented at the 2010 International Conference on Emerging Security Technologies, pp. 71–76 (2010)
9. Poulos, M., Rangoussi, M., Chrissikopoulos, V., Evangelou, A.: Person identification based on parametric processing of the EEG. Presented at the 6th IEEE International Conference on Electronics, Circuits and Systems. Proceedings of ICECS 1999, vol. 1, pp. 283–286 (1999)
10. Marcel, S., Millan, J.R.: Person Authentication Using Brainwaves (EEG) and Maximum A Posteriori Model Adaptation. IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), 743–752 (2007)
11. Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., Flor, H.: A spelling device for the paralysed. Nature **398**(6725), 297–298 (1999)
12. Bensch, M., Karim, A.A., Mellinger, J., Hinterberger, T., Tangermann, M., Bogdan, M., Rosenstiel, W., Birbaumer, N.: Nessi: An EEG-Controlled Web Browser for Severely Paralyzed Patients. Comput. Intell. Neurosci. **2007**, 1–5 (2007)
13. Finke, A., Lenhardt, A., Ritter, H.: The MindGame: A P300-based brain–computer interface game. Neural Netw. **22**(9), 1329–1333 (2009)
14. Lopez, M.A., Pomares, H., Damas, M., Prieto, A.G., de la Plaza Hernandez, E.M.: Use of kohonen maps as feature selector for selective attention brain-computer interfaces. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2007. LNCS, vol. 4527, pp. 407–415. Springer, Heidelberg (2007)
15. López, M., Pomares, H., Damas, M., Madrid, E., Prieto, A.G., Pelayo, F.J., de la Plaza Hernández, E.M.: Use of ANNs as classifiers for selective attention brain-computer interfaces. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 956–963. Springer, Heidelberg (2007)
16. Volosyak, I., Moor, A., Gräser, A.: A dictionary-driven SSVEP speller with a modified graphical user interface. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part I. LNCS, vol. 6691, pp. 353–361. Springer, Heidelberg (2011)
17. Volosyak, I.: SSVEP-based Bremen–BCI interface—boosting information transfer rates. J. Neural Eng. **8**(3), 036020 (2011)

18. Lee, H.J., Kim, H.S., Park, K.S.: A study on the reproducibility of biometric authentication based on electroencephalogram (EEG), pp. 13–16 (2013)
19. Palaniappan, R., Ravi, K.V.R.: A new method to identify individuals using signals from the brain, vol. 3, pp. 1442–1445 (2003)
20. Palaniappan, R., Ravi, K.V.R.: Improving visual evoked potential feature classification for person recognition using PCA and normalization. Pattern Recognit. Lett. **27**(7), 726–733 (2006)
21. Pham, T., Ma, W., Tran, D., Nguyen, P., Phung, D.: Multi-factor EEG-based user authentication, pp. 4029–4034 (2014)
22. Cecotti, H., Rivet, B., Congedo, M., Jutten, C., Bertrand, O., Maby, E., Mattout, J.: A robust sensor-selection method for P300 brain–computer interfaces. J. Neural Eng. **8**(1), 016001 (2011)
23. Palaniappan, R., Gosalia, J., Revett, K., Samraj, A.: PIN generation using single channel EEG biometric. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (eds.) ACC 2011, Part IV. CCIS, vol. 193, pp. 378–385. Springer, Heidelberg (2011)
24. Thorpe, J., van Oorschot, P.C., Somayaji, A.: Pass-thoughts: authenticating with our minds, p. 45 (2005)
25. Paranjape, R.B., Mahovsky, J., Benedicenti, L., Koles, Z.: The electroencephalogram as a biometric, vol. 2, pp. 1363–1366 (2001)
26. Klonovs, J., Petersen, C.K., Olesen, H., Hammershoj, A.: ID Proof on the Go: Development of a Mobile EEG-Based Biometric Authentication System. IEEE Veh. Technol. Mag. **8**(1), 81–89 (2013)
27. Hema, C.R., Osman, A.A.: Single trial analysis on EEG signatures to identify individuals, pp. 1–3 (2010)
28. Rizwan-i-Haque, I., Khan, M.F., Saleem, M., Rao, N.I.: Network weight adjustment in a fractional fourier transform based multi-channel brain computer interface for person authentication, pp. 900–905 (2012)
29. Riera, A., Soria-Frisch, A., Caparrini, M., Grau, C., Ruffini, G.: Unobtrusive Biometric System Based on Electroencephalogram Analysis. EURASIP J. Adv. Signal Process. **2008**(1), 143728 (2008)
30. Safont, G., Salazar, A., Soriano, A., Vergara, L.: Combination of multiple detectors for EEG based biometric identification/authentication, pp. 230–236 (2012)
31. The Smart Grid Interoperability Panel–Smart Grid Cybersecurity Committee: Guidelines for smart grid cybersecurity. National Institute of Standards and Technology, NIST IR 7628r1 (September 2014)
32. Bhattasali, T., Saeed, K., Chaki, N., Chaki, R.: A survey of security and privacy issues for biometrics based remote authentication in cloud. In: Saeed, K., Snášel, V. (eds.) CISIM 2014. LNCS, vol. 8838, pp. 112–121. Springer, Heidelberg (2014)
33. Kalra, S., Lamba, A.: A survey on multimodal biometric. Int. J. Comput. Sci. Inf. Technol. **5**(2), 2148–2151 (2014)
34. Verma, D., Dubey, S.: A Survey on Biometric Authentication Techniques Using Palm Vein Feature. J. Glob. Res. Comput. Sci. **5**(8), 5–8 (2014)
35. Hakvoort, G., Gürkök, H., Plass-Oude Bos, D., Obbink, M., Poel, M.: Measuring immersion and affect in a brain-computer interface game. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011, Part I. LNCS, vol. 6946, pp. 115–128. Springer, Heidelberg (2011)
36. Lopez-Gordo, M.A., Pelayo, F., Prieto, A.: A high performance SSVEP-BCI without gazing. In: The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, pp. 193–197 (2010)

37. Wang, H., Zhang, C., Shi, T., Wang, F., Ma, S.: Real-Time EEG-Based Detection of Fatigue Driving Danger for Accident Prediction. Int. J. Neural Syst., p. 1550002 (November 2014)
38. Gao, X., Xu, D., Cheng, M., Gao, S.: A BCI-Based Environmental Controller for the Motion-Disabled. IEEE Trans. Neural Syst. Rehabil. Eng. **11**, 137–140 (2003)
39. Spüler, M., Rosenstiel, W., Bogdan, M.: Online Adaptation of a c-VEP Brain-Computer Interface(BCI) Based on Error-Related Potentials and Unsupervised Learning. PLoS ONE **7**(12), e51077 (2012)
40. Lopez-Gordo, M.A., Ron-Angevin, R., Pelayo Valle, F.: Auditory brain-computer interfaces for complete locked-in patients. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part I. LNCS, vol. 6691, pp. 378–385. Springer, Heidelberg (2011)
41. Kübler, A., Neumann, N., Kaiser, J., Kotchoubey, B., Hinterberger, T., Birbaumer, N.: Brain-computer communication: Self-regulation of slow cortical potentials for verbal communication. Arch. Phys. Med. Rehabil. **82**(11), 1533–1539 (2001)
42. Iversen, I.H., Ghanayim, N., Kübler, A., Neumann, N., Birbaumer, N., Kaiser, J.: A brain–computer interface tool to assess cognitive functions in completely paralyzed patients with amyotrophic lateral sclerosis. Clin. Neurophysiol. **119**(10), 2214–2223 (2008)
43. Lopez-Gordo, M.A., Pelayo, F., Prieto, A., Fernandez, E.: An Auditory Brain-Computer Interface with Accuracy Prediction. Int. J. Neural Syst. **22**(3), 1–14 (2012)
44. Lopez-Gordo, M.A., Pelayo, F.: A Binary Phase-Shift Keying Receiver for the Detection of Attention to Human Speech. Int. J. Neural Syst., p. 130418190845004 (March 2013)
45. Ron-Angevin, R., Varona-Moya, S., da Silva-Sauer, L., Carrion-Robles, T.: A brain–computer interface speller with a reduced matrix: a case study in a patient with amyotrophic lateral sclerosis. Presented at the COGNITIVE 2014: The Sixth International Conference on Advanced Cognitive Technologies and Applications (2014)
46. Ron-Angevin, R., Lopez, M.A., Pelayo, F.: The training issue in brain-computer interface: a multi-disciplinary field. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 666–673. Springer, Heidelberg (2009)

A Label-Aided Filter Method for Multi-objective Feature Selection in EEG Classification for BCI

Pedro Martín-Smith[1], Julio Ortega[1(✉)], Javier Asensio-Cubero[2], John Q. Gan[2], and Andrés Ortiz[3]

[1] Department of Computer Architecture and Technology, CITIC, University of Granada, Granada, Spain
{pmartin,jortega}@ugr.es
[2] School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
{jasens,jqgan}@essex.ac.uk
[3] Department of Communications Engineering, University of Malaga, Malaga, Spain
aortiz@ic.uma.es

Abstract. This paper proposes and evaluates a filter approach for evolutionary multi-objective feature selection in classification problems with a large number of features. Such classification problems frequently appear in many bioinformatics applications where the number of patterns is smaller than the number of features and thus the curse of dimensionality problem exists. The main contribution of this paper is proposing a set of label-aided utility functions that allows the effective search of the most adequate subset of features through an evolutionary multi-objective optimization scheme. The experimental results have been obtained in a brain-computer interface (BCI) classification task based on LDA classifiers, where the properties of multi-resolution analysis (MRA) for signal analysis in temporal and spectral domains have been used to extract the features from EEG signals. The results from the proposed filter method demonstrate some advantages such as less time consumption and better generalization capabilities with respect to some wrapper-based multi-objective feature selection alternatives.

Keywords: Brain-Computer Interfaces (BCI) · Filter methods · Feature selection · Multi-objective optimization · Multi-Resolution Analysis (MRA)

1 Introduction

Many high-dimensional pattern classification or modeling tasks require feature selection techniques in order to remove redundant, noisy-dominated, or irrelevant inputs. In particular, dimensionality reduction is very important to improve the accuracy and interpretability of the classifiers when the number of features is too large compared to the number of available training patterns, which is known as the curse of dimensionality. In bioinformatics, the high dimensional character of the involved modeling problems is frequent thus making feature selection an important issue. Saeys et al. [1]

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 133–144, 2015.
DOI: 10.1007/978-3-319-19258-1_12

reviewed feature selection techniques used in bioinformatics along with analyses and references of feature selection in different bioinformatics applications such as sequence analysis, microarray analysis, and mass spectra analysis. In [2], feature selection in high-dimensional feature spaces with small pattern samples is considered in the search of sets of genes whose expression levels can serve as features for diagnosis or prognosis. Dimensionality reduction in the input patterns has been also applied to EEG (Electroencephalogram) classification for recognizing epileptiform patterns [3]. In EEG classification for BCI applications [4], the following problems make feature selection an important issue: (1) the presence of noise or outliers in the features; (2) the need to represent time information in the features; (3) the non-stationarity of EEG signals; and (4) the small number of patterns (EEGs) available for training. This circumstance results in the so-called curse-of-dimensionality problem as the number of patterns needed to properly define the different classes increases exponentially with the dimension of the feature vectors (at least from five to ten times as many training samples per class as the dimension [5]).

This paper deals with multi-objective feature selection in supervised classification problems. To evaluate the proposed method, motor imagery (MI) based BCI tasks have been used as benchmark EEG classification problems where multi-resolution analysis (MRA) has been applied for feature extraction [6]. An MRA system applies a sequence of successive approximation spaces that satisfy a series of constraints to reach a description as close as possible to the target signal [7], and thus it is useful whenever the target signal presents different characteristics in the successive approximation spaces. MI is a BCI paradigm that uses the series of amplifications and attenuations of short duration occasioned by limb movement imagination, the so called event related desynchronization (ERD) and event related synchronization (ERS) [8]. The task of ERD/ERS analysis is complex because ERD/ERS patterns are weak and noisy and occur at different locations of the cortex, at different instants within a trial, and in different frequency bands. Moreover, there is no consistency in the patterns among subjects, and the patterns can even change within a session for the same subject, which may lead to high-dimensional patterns making the number of available patterns to conduct ERD/ERS analysis significantly less than the number of features. This constitutes a good scenario for evaluating the multi-objective feature selection approaches.

This paper is organized as follows. Section 2 introduces feature selection as a multi-objective optimization problem, while the filter approach for evolutionary multi-objective feature selection that constitutes the main contribution of this paper is presented in Section 3. Section 4 describes the data acquisition and feature extraction, by multi-resolution analysis (MRA), of EEG for the BCI tasks considered as benchmarks. The previously proposed feature selection procedures used to compare with our proposal are also described in Section 4. Finally, the experimental results obtained in this application are shown in Section 5, and the conclusions are given in Section 6.

2 Multi-objective Feature Selection

In this paper, feature selection is accomplished through an approach which can be regarded as searching for features optimizing some cost functions. These cost functions evaluate the utility of the selected features for correct classification of multi-class patterns. As there are more than one cost functions to be optimized simultaneously, a multi-objective optimization procedure should be applied.

A multi-objective optimization problem can be defined as finding a vector of decision variables $\mathbf{x}=[x_1,x_2,...,x_n]\in R^n$ that satisfies a restriction set, e.g., $g(\mathbf{x})\leq 0$, $h(\mathbf{x})=0$, and optimizes a function vector $\mathbf{f}(\mathbf{x})$, whose scalar values $(f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_m(\mathbf{x}))$ represent the objectives of the optimization. As these objectives are usually in conflict, instead of providing only one optimal solution, the procedures applied to multi-objective optimization should obtain a set of *non-dominated* solutions, known as Pareto optimal solutions, from which a decision agent will choose the most convenient solution in specific circumstances. These Pareto optimal solutions are optimal in the sense that in the corresponding hyper-area known as *Pareto front*, no solution is worse than the others when all the objectives are taken into account.

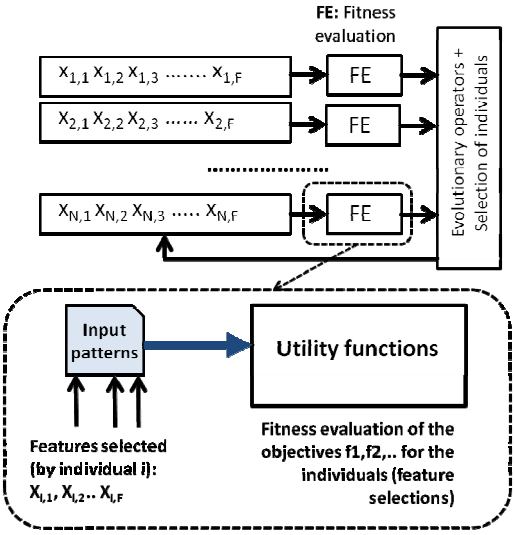


Fig. 1. Scheme of an evolutionary multi-objective optimization approach for feature selection

Figure 1 provides a scheme of multi-objective optimization for feature selection. Each individual of the population encodes the selected features of the input patterns. To define the utility functions used as fitness functions in the evolutionary approach of Figure 1, *wrapper*, *embedded*, *filter*, or *hybrid* alternatives have been proposed. In [9] a revision of previous work done on the different feature selection methods is provided. The *wrapper* and *embedded* approaches are classifier-dependent and use the classifier performance to evaluate the utility of the selected features. Instead, *filter* approaches take into account characteristics of the training patterns such as Euclidean

distances, correlation-based functions, consistency-based functions, information measures, etc., to define the utility functions independent of any classifier. This way, a filter approach for feature selection does not require training the classifier used in the problem at hand.

The steps of the evolutionary algorithm used in this paper for multi-objective optimization in Figure 1 correspond to those of the NSGA-II algorithm [10], with specific individual codification and genetic operators implemented for the application at hand. In NSGA-II the fitness values of the individuals in the population are sorted according to the different fronts of non-dominated individuals (non-domination levels) where they belong, while the diversity among individuals in the same non-dominated front is also preserved. This way, NSGA-II uses a fast procedure of $O(MN^2)$ complexity (M is the number of objectives and N the number of individuals in the population), and the storage requirements grow as $O(N^2)$ in order to sort the population according to the different levels of non-dominance. To maintain the diversity among solutions with the same non-dominance level, NSGA-II estimates the density of solutions surrounding a given solution through the average distance of the nearest neighbour solutions on either side of the considered solution for each dimension (objective) of the front. The density of solutions or crowding distance can be computed with complexity O(MNlogN) as it implies to sort, at most, N individuals along the dimensions corresponding to M objectives. This crowding distance avoids the need of setting a sharing parameter. Once each individual in the population has a non-domination rank and a crowding distance, a selection process chooses solutions with lower non-domination ranks and solutions with higher crowding distances (less crowded regions) in case of similar non-domination ranks. Thus, the complexity of NSGA-II is determined by the population sorting step, i.e., $O(MN^2)$.

3 A Label-Aided Filter Approach for Evolutionary Multi-objective Feature Selection

As feature selection can be for either supervised or unsupervised learning, different utility functions can be considered [11]. With respect to supervised classifiers, multi-objective feature selection procedures often take into account the number of features and the performance of the classifier [12, 13]. There are some studies focusing on feature selection for unsupervised learning [11, 14, 15]. Although the labels for training and testing patterns are usually available in BCI datasets, they are not always accurate and thus clustering would be useful to model the intrinsic pattern structure before training the classifier. Moreover, it is possible to follow the same strategy as in semi-supervised clustering algorithms, whose key idea is to take advantage of some prior knowledge to improve the performance of clustering [16, 17]. In [18], a consistency-based criterion is applied to select relevant features: in a set of consistent features, no two patterns with the same values in all the features could have different class labels. This kind of principle has been followed in this paper to derive our filter method for feature selection, where it is supposed that close patterns in the space of selected features should belong to similar classes.

```
Input
Patterns={P1,..,Pn} with Pi=(pi,1,..,pi,r) (i=1,..,n)
Labels={Li} with Li∈{1,..,NC} being the label of pattern Pi
NC is the number of classes
S(m)=0 (m=1,..,NC) are temporal variables to evaluate label
coincidences/differences among couples of patterns
SS(m,k)=0 (m=1,..,NC; k=1,..,Q) are coincidence measures and Q
is the number of possible couples of patterns

Output
{F(1),..,F(NC)} cost functions values (as functions as labels)
_____________________________________________________________
01    Compute Di,j=d(Pi,Pj) (i,j=1,..,n, i<j);
      // Euclidean distance between patterns i and j
02    Δsort ={Δk(k=1,...,Q)} (Q=n(n-1)/2))
      // with Δk=(Pσ1(k),Pσ2(k),d(Pσ1(k),Pσ2(k)))
      // such that σ1(k)<σ2(k) and d(Pσ1(k),Pσ2(k))≤d(Pσ1(k+1),Pσ2(k+1))
      // for k=1,..,Q-1
03    for k=1 to W // (W ≤ Q)
      {
04          if (Lσ1(k)=Lσ2(k)) then
05                S(Lσ1(k))= S(Lσ1(k))+1; // coincidence
06          else
07                S(Lσ1(k))= S(Lσ1(k))-0.5); // different classes
08                S(Lσ2(k))= S(Lσ2(k))-0.5);
09          end
10          for m=1 to NC SS(m,k)=S(m);
      }
11    for m=1 to NC
      {
12          F(m)=-averagek(SS(m,k));
      }
```

Fig. 2. Utility function evaluation by the proposed filter approach for multi-objective feature selection

Figure 2 shows the pseudo-code of the label-aided filter procedure proposed to evaluate the fitness of individuals of the population in the multi-objective approach for feature selection. As can be seen, a utility function is defined for each class and is used as a cost (or fitness) function in the multi-objective optimization procedure for feature selection. Given n training patterns, $P_i,\ldots,P_n$, and their corresponding labels $L_i,..,L_n$, corresponding to NC different classes, the first step (step 01) in Figure 2 is devoted to compute the Euclidean distances among all the pairs of patterns. Then (step 02) the different pairs of patterns are sorted (in increasing order) according to their Euclidean distances. After that, the first W sorted elements in Δ_{sort}, Δ_k (k=1,..,W) are considered. As it is shown in Figure 2, Δ_k includes two patterns and their mutual distance, and σ1(k) and σ2(k) are the indexes of, respectively, the first and second patterns in Δ_k. These W couples of nearest patterns are used to compute a set of coincidence measures S(m, k) (m=1,..,NC, k=1,..,W) in steps 03 to 10. If two consecutive patterns correspond to the same label, the coincidence measure corresponding to this label is incremented by one as it is supposed that closer patterns belonging to the same cluster should be labelled as patterns of the same class. On the contrary, if closer patterns do not belong to the same class, the coincidence measure is decreased by 0.5

(one divided by two, as we distribute the cost decrement between a couple of labels). Finally, in steps 11 and 12, the cost or utility functions F(1) to F(NC) are computed as $F(m)=-\sum_k SS(m,k)/W$, m=1,..,NC. This way, the utility functions F(1) to F(NC) allow us to evaluate the consistency between couples of patterns with low mutual distances and high probabilities of label coincidence for patterns in the corresponding couple. Figure 3 provides an idea about the behaviour of the cost functions. As more coincidences arise in close patterns, the curve corresponding to SS(i,k) rises faster and the size of area drawn in the figure usually increases.

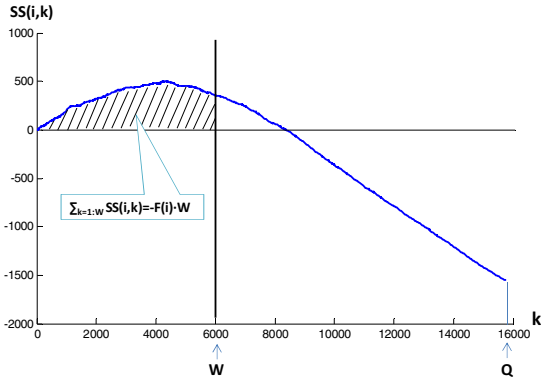


Fig. 3. Utility function and coincidence measures

4 Application of the Proposed Method to BCI Feature Selection

In this section, we describe the dataset and the previously proposed procedures used to evaluate our filter-based multi-objective feature selection approach.

The dataset was obtained from the BCI Laboratory at the University of Essex, and includes patterns that correspond to three different classes of imagined movements (right hand, left hand, and feet) from 10 subjects with ages from 24 to 50 (58% female, 50% naïve to BCI). It was recorded with a sampling frequency of 256 Hz during four different runs and includes a total of 120 trials for each class and subject. More details about this dataset can be found in [6].

Each pattern was obtained from an EEG trial by the feature extraction procedure based on the MRA described in [6]. Thus, each signal obtained from each electrode contains several segments to which a set of wavelet detail and approximation coefficients are assigned. If there are S segments, E electrodes, and L levels of wavelets, each pattern is characterized by 2×S×E×L sets of coefficients (the number of coefficients in each level set depends on the level). In the Essex BCI dataset, S=20 segments, E=15 electrodes, and L=6 levels, therefore there are 3600 sets of coefficients in total, with from 4 to 128 coefficients in each set are used to characterize each pattern (a total of 151200 coefficients). Figure 4 shows how the pattern features are generated. Taking into account that the number of training patterns for each subject is approximately 180, it is clear that an efficient procedure for feature selection is required.

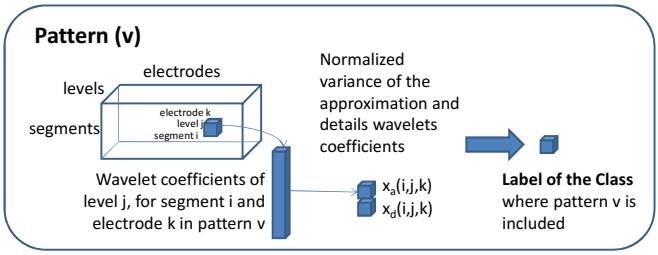


Fig. 4. Characterization of an EEG signal (pattern) in [6]

In [6] a simple approach to reduce the number of coefficients is applied, in which only one coefficient is assigned to each electrode and each level of approximation and detail. This coefficient is obtained by computing the second moment of the coefficient distribution (variance) and normalizing the value between 0 and 1. This way, the number of coefficients for a given pattern is 2×S×E×L.

An approach to alleviate the curse of dimensionality without a feature selection process is to use a set of LDA (linear discriminant analysis) classifiers and to adopt the majority voting of all the LDA outputs as the final classification output [6]. Thus, a set of 2×S×L LDA classifiers with the number of inputs equaling the number of electrodes, E, is required. This alternative will be called OPT0 in what follows.

Two more options of multi-objective feature selection, OPT1 and OPT2 [20], have been also used to compare with the filter approach proposed in this paper. These two approaches implement wrapper procedures to evaluate the cost functions of the multi-objective optimization procedure. Option OPT1 simply uses one LDA classifier and searches for the best feature subset as its input, and option OPT2 selects features using 2×S×L LDA classifiers, with E inputs in each LDA as in OPT0.

Both OPT1 and OPT2 are wrapper approaches that can take advantage of the performance of the LDA classifier once it has been trained with the corresponding feature selection. In this case, to characterize the performance of the classifier while it has been trained or adjusted for a given set of features (an individual of the population), it has been considered that it is important not only to take into account the accuracy obtained for the training set, but also to its behaviour for unseen instances, i.e., its generalization capabilities. Thus, two cost functions are proposed to evaluate the feature selection procedures OPT1 and OPT2. The first one is the kappa index [19], which provides an accurate description of the classifier performance. It can be considered even better than the classification ratio as it takes into account the per class error distribution. The other cost function estimates aspects such as the generalization capability or the classifier overfitting. In our case, 10-fold cross-validation analysis to the training patterns has been taken into account to define the second cost function.

5 Experimental Results

This section presents the experimental results obtained by the evolutionary multi-objective feature selection approaches OPT0, OPT1 and OPT2, and the filter approach described in Section 3 (it is called FOPT1 hereafter). The procedures have been implemented in MATLAB®, and have been executed in a computer with processor P8700 at 2.53 GHz and 4GB of RAM. The experiments have been conducted by using the dataset recorded in the BCI Laboratory at the University of Essex. For each subject, there is one data file named x# with data recorded in two runs for training and another data file named xe# with data recorded in two runs for evaluation. Each data file contains about 180 labelled patterns with data from 20 segments (S=20), six levels (L=6) of approximation or detail coefficients (a/d=2), and 15 electrodes (E=15). The labels correspond to three imagined movements of right hand, left hand, and feet. According to this, in OPT0, S×L×2=240 LDAs with 15 inputs are considered for voting the class to which the corresponding pattern belongs. The input from a given electrode is the normalized second moment of the wavelet coefficients of the signal in this electrode, for the s-th segment, the l-th level, and approximation/detail type of the corresponding LDA. In OPT0 there is no feature selection. OPT1 applies a multi-objective feature selection procedure on S×L×E×2 features corresponding to the possible segments, levels, variances of the approximation and detail coefficients, and electrodes. This way, in the BCI dataset of the University of Essex there are S×L×E×2=20×6×15×2=3600 possible features for selection in OPT1 and in the proposed filter approach FOPT1. In the case of OPT2, the dimension of the search space for feature selection is 20×6×2=240.

For OPT1 and FOPT1, as the number of features available is very large, we have not codified the individuals by using binary components, with each component corresponding to a possible feature. In our approach, each individual is codified by a set of vectors, with each vector corresponding to one of the features included in the selection codified by the individual. The components of the vector correspond to the dimensions that characterize each input pattern. For example, in the case described in Figure 4, each feature is codified by a vector with four components: segment, level, electrode, analysis/detail wavelet component. Each component will have an integer value between 1 and S for the segment component, between 1 and L for the level component, between 1 and E for the electrode component, and 0 or 1 for the analysis/detail component. This strategy provides efficient codification and allows to extract information about the characteristics of the feature selection (segments, levels, electrodes, or coefficients more frequently selected, etc.).

With respect to the implemented evolutionary multi-objective algorithm, two different operators can be applied to the individuals of the population. The first one is a crossover operator that randomly selects two parents and a subset (also randomly selected) of features for each parent. These subsets are interchanged by the parents. The mutation operator applies changes to a subset of features randomly chosen among the features codified by the individual to be mutated, which is also randomly selected. The changes in the values of the components of the feature to be mutated are normally distributed with means and deviations that change with generations. After some

experiments, we have used a crossover probability of 0.9, a mutation probability equal to 1/n (n is the number of decision variables), and distribution indices for crossover and mutation operators equal to 20.

The evolutionary multi-objective feature selection procedures have been executed with different populations and generations to determine the minimum numbers that provide competitive results compared with OPT0. OPT1 and OPT2 have been executed by using 50 individuals and 50 generations so that the amount of searching work in all the approaches is similar. The procedure FOPT1 has been executed with populations of 200 individuals, 300 generations, and W=Q (number of pairs of patterns). Executions with different number of iterations and individuals could change the comparison results among the different approaches. We will address this issue at the end of this section.

We have used simulated binary crossover with a crossover probability of 0.5, a mutation probability of 0.5, and distribution index of 20 for crossover and mutation operators. It is worth mentioning that no work on tuning the parameters of the evolutionary multi-objective feature selection options to optimize their behavior has been considered, as our aim here is to analyze whether multi-objective optimization based on a filter procedure is able to provide some improvements on MRA approaches for BCI.

Tables 1 and 2 compare the Kappa indexes obtained by the 4 approaches (OPT0, OPT1, OPT2, and FOPT1). The columns labeled as "Kappa index (x#)" in Table 1 correspond to the Kappa index values obtained by the trained classifier when it was evaluated by using the same training patterns. This information is only used to give an idea of the difficulty of the problem. The procedures are evaluated by using the test patterns, in Table 2. The columns labeled as "Kappa index (xe#)" in Table 2 provide the Kappa index values obtained by the 4 approaches when the testing patterns were used to evaluate the classifier performance for each subject.

Table 1. Comparison of different feature selection and classification methods for the University of Essex BCI data files (Kappa values evaluated with the training patterns)

	OPT0	OPT1	OPT2	FOPT1
Subject	Kappa index (x#)	Kappa index (x# mean,std)	Kappa index (x# mean,std)	Kappa index (x# mean,std)
101	0.790	0.738±0.022*	**0.828±0.011^**	0.744±0.014*
102	**0.857**	0.747±0.017*	**0.855±0.011**	0.767±0.023*
103	**0.757**	0.665±0.012*	**0.747±0.013**	0.659±0.020*
104	0.899	0.819±0.013*	**0.902±0.010**	0.844±0.018*
105	**0.757**	0.646±0.023*	**0.751±0.013**	0.628±0.023*
106	0.774	0.604±0.023*	**0.776±0.014**	0.604±0.015*
107	0.857	0.816±0.018*	**0.880±0.011^**	0.820±0.020*
108	**0.774**	0.508±0.021*	0.745±0.023*	0.500±0.018*
109	**0.790**	0.597±0.021*	0.770±0.015*	0.595±0.023*
110	0.883	0.821±0.016*	0.877±0.026	0.831±0.018*

The results shown in Table 1 and Table 2 for each subject correspond to the average (mean and standard deviation) over 15 executions for OPT1 and OPT2, and 10 executions for FOPT1. Statistical analysis has been conducted by applying a Kolmogorov-Smirnov

test first to determine whether the obtained values of the Kappa index follow a normal distribution or not. If the experimental results do not have normal distribution, a non-parametric Kruskal-Wallis test has been used to compare the means of the different algorithms. A confidence level of 95% has been considered in the statistical tests. If the mean values of the Kappa index for the different options are statistically significantly different from those in OPT0 according to the Kruskal-Wallis test, they are marked with either ^ or *, with ^ indicating performance improvement and * indicating performance loss.

Table 2. Comparison of different feature selection and classification methods for the University of Essex BCI data files (Kappa values evaluated with the test patterns)

	OPT0	OPT1	OPT2	FOPT1
Subject	Kappa index (xe#)	Kappa index (xe# mean,std)	Kappa index (xe# mean,std)	Kappa index (xe# mean,std)
101	**0.438**	0.393±0.046*	**0.437±0.033**	**0.468±0.028^**
102	**0.455**	0.302±0.074*	**0.429±0.023**	0.406±0.025*
103	0.279	0.249±0.046	0.325±0.017^	**0.354±0.029^**
104	**0.564**	**0.510±0.056**	**0.545±0.035**	**0.639±0.045^**
105	**0.287**	0.191±0.040*	**0.240±0.031**	**0.248±0.041**
106	**0.321**	0.193±0.070*	**0.319±0.028**	0.271±0.028*
107	0.631	0.560±0.041*	**0.634±0.019**	**0.636±0.018**
108	**0.254**	0.088±0.036*	0.184±0.027*	0.190±0.028*
109	**0.388**	0.207±0.071*	0.333±0.026*	0.318±0.041*
110	**0.648**	0.450±0.036*	0.605±0.041*	0.559±0.022*

With respect to the execution time required by each approach, the mean execution time is 4533±45s for OPT1, 13353±1031s for OPT2, and 11598±67s for FOPT1. Taking into account these differences in the running time, it is possible to argue that the comparison among different approaches may not be fair as (specially) OPT1 could probably achieve better results when more generations (with more individuals) are adopted with execution time similar to that required by OPT2 and FOPT1. We have made experiments for two subjects (x104 and x107) with OPT1 using a population of 100 individuals and 60 generations. In these cases OPT1 requires a mean execution time of 11042±162s, which is in a similar order as the one required by FOPT1. With respect to the results in these experiments, 0.515±0.047 is obtained for x104 and 0.580±0.047 for x107. In both cases, the results obtained by our procedure FOPT1 are still better.

6 Conclusions

This paper proposes a filter approach for feature selection that evaluates a cost or utility function for each class in the classification problem. Then, a multi-objective optimization procedure can be used to determine a set of non-dominated feature selections among which the user can determine the best one according to his/her requirements.

The proposed method has been compared with a procedure [6] that copes with the curse of dimensionality based on the composition of multiple classifiers, where each classifier receives only a subset of pattern components (features) in such a way that

the number of patterns is much higher than the number of features used as inputs. The problem with this kind of approaches is that the number of classifiers to train and to accomplish the classification is usually very high and the need to compute such a high number of features and to train a lot of classifiers could be a significant drawback to satisfy real time requirements of many applications.

The multi-objective filter approach has also been compared with other two previously proposed multi-objective feature selection procedures based on a wrapper method [20]. They evaluate the LDA classification performance according to two cost functions corresponding, respectively to the classifier accuracy and its generalization capability.

The experimental results show that evolutionary multi-objective feature selection is able to provide classification performance similar to that of using all the possible LDAs with all the possible feature inputs (OPT0). Thus the proposed approaches lead to simpler classification procedures with fewer features. The proposed filter method, FOPT1, provides better performances than the wrapper approach OPT1, which uses the same LDA structure as FOPT1. FOPT1 is also able to outperform OPT0 and OPT2 in some cases. Moreover, the generalization capability seems to be better for FOPT1 than for both OPT1 and OPT2 as the differences among the Kappa values in Tables 1 and 2 are lower for FOPT1.

As future work, we will consider the analysis of the characteristics of the features selected for obtaining some knowledge about important electrodes and segments. Moreover, we will continue our studies on clustering validation indexes for both wrapper and filter feature selection methods.

Acknowledgements. This work was partly funded by grant TIN2012-32039 (Spanish "Ministerio de Economía y Competitividad" and FEDER funds). The authors would like to thank the anonymous reviewers for their useful comments and suggestions.

References

1. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**(19), 2507–2517 (2007)
2. Sima, C., Dougherty, E.: What should be expected from feature selection in small-sample settings. Bioinformatics **22**, 2430–2436 (2006)
3. Acır, N., Güzeliş, C.: An application of support vector machine in bioinformatics: automated recognition of epileptiform patterns in EEG using SVM classifier designed by a perturbation method. In: Yakhno, T. (ed.) ADVIS 2004. LNCS, vol. 3261, pp. 462–471. Springer, Heidelberg (2004)
4. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A review of classification algorithms for EEG-based brain-computer interfaces. Journal of Neural Engineering, **4** (2007)
5. Raudys, S.J., Jain, A.K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(3), 252–264 (1991)

6. Asensio-Cubero, J., Gan, J.Q. Palaniappan, R.: Multiresolution analysis over simple graphs for brain computer interfaces. Journal of Neural Engineering, **10**(4) (2013). doi:10.1088/1741-2560/10/4/046014
7. Daubechies, I.: Ten Lectures on Wavelets. SIAM, Philadelphia (2006)
8. Pfurtscheller, G., Lopes da Silva, F.H.: Event-related EEG/MEG synchronization and desynchronization: Basic principles. Clinical Neurophysiology **110**(11), 1842–1857 (1999)
9. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. Knowl. Inf. Syst. **34**, 483–519 (2013)
10. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A fast elitist Non-dominated Sorting Genetic Algorithms for multi-objective optimisation: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
11. Handl, J., Knowles, J.: Feature selection in unsupervised learning via multi-objective optimization. Intl. Journal of Computational Intelligence Research **2**(3), 217–238 (2006)
12. Emmanouilidis, C., Hunter, A., MacIntyre, J.: A multi-objective evolutionary setting for feature selection and a commonality-based crossover operator. In: Proc. of the 2000 Congress on Evolutionary Computation, pp. 309–316. IEEE Press, New York (2000)
13. Oliveira, L.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. International Journal of Pattern Recognition and Artificial Intelligence **17**(6), 903–929 (2003)
14. Kim, Y., Street, W.N., Menczer, F.: Evolutionary model selection in unsupervised learning. Intelligent Data Analysis **6**(6), 531–556 (2002)
15. Morita, M., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In: Proc. of the Seventh International Conference on Document Analysis and Recognition, pp. 666–671. IEEE Press, New York (2003)
16. Gan, H., Sang, N., Huang, R., Tong, X., Dan, Z.: Using clustering analysis to improve semi-supervised classification. Neurocomputing **101**, 290–298 (2013)
17. Basu, S., Banerjee, A., Rooney, R.J.: Semi-supervised clustering by seeding. In: Proc. of the 19th International Conference on Machine Learning, pp. 11–18 (2003)
18. Dash, M., Liu, H.: Consistency-based search in feature selection. Artificial Intelligence **151**, 155–176 (2003)
19. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychological Meas. **20**, 37–46 (1960)
20. Ortega, J., Asensio-Cubero, J., Gan, J.Q., Ortiz, A.: Evolutionary multiobjective feature selection in multiresolution analysis for BCI. In: Ortuño, F., Rojas, I. (eds.) IWBBIO 2015, Part I. LNCS, vol. 9043, pp. 347–359. Springer, Heidelberg (2015)

Multi-Robot Systems: Applications and Theory (MRSAT)

A First Step Toward a Possibilistic Swarm Multi-robot Task Allocation

José Guerrero(✉), Óscar Valero, and Gabriel Oliver

Mathematical and Computer Science Department,
University of the Balearic Islands, Cra. de Valldemossa Km 7.5.,
07122 Palma de Mallorca, Spain
{jose.guerrero,o.valero,goliver}@uib.es

Abstract. The task allocation problem is one of the main issues in multi-robot systems. Typical ways to address this problem are based on Swarm Intelligence. One of them is the so-called Response Threshold Method. In the aforementioned method every robot has associated a task response threshold and a task stimuli in such a way that the robot's probability of executing a certain task depends on both factors. On of the advantage of the aforesaid method is given by the fact that the original problem is treated from a distributed mode which, at the same time, means a very low computational requirements. However, the Response Threshold Method cannot be extended in a natural way to allocate more than two tasks when the theoretical basis is provided by probability theory. Motivated by this fact, this paper leaves the probabilistic approach to the problem and takes a first step towards a possibilistic theoretical approach in order to treat successfully the multi-robot task allocation problem when more than two tasks must be performed. As an example of application, an scenario where each robot task stimuli only depends on the distance between tasks is studied and the convergence of the system to an stable state is shown.

Keywords: Multi-robot · Possibility theory · Swarm intelligence · Task allocation

1 Introduction

Systems with two or more mobile robots (multi-robot-systems) can perform tasks that with only one robot would be impossible to carry out or would take a lot of time. Moreover, such systems are more robust, scalable and flexible than those with only one robot. A lot of new challenges and problems must be solved before taking advantage of the potential benefits of the multi-robot systems. Among all possible issues that arise in a natural way in multi-robot systems, this paper focuses on the problem commonly referred to as "Multi-robot Task Allocation" problem (MRTA for short) which consists of selecting the best robot to execute each of the tasks that must be performed. A lot of research has been done to solve the aforementioned problem in the last years. Concretely, many efforts have

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 147–158, 2015.
DOI: 10.1007/978-3-319-19258-1_13

been made to propose suitable methods based on auction and swarm strategies for task allocation.

Auction paradigms [7,10] are based on explicit communication protocols between robots: when a robot, called auctioneer, finds or generates a new task sends a message to announce it before starting the execution. The other robots, called bidders, send to the auctioneer a value, called bid, that indicates how suitable is itself for executing the new task. Once the auctioneer has received all the bids, it selects the robot with the highest bid for the task. Auction methods, and negotiation paradigms in general, provide better solutions than swarm approaches. Nevertheless, the communication requirement in such methods can become a drawback.

Swarm methods are inspired by insect colonies behavior, where the cooperative behavior emerges from the interaction of very simple behaviors running on each robot without any communication protocol [15]. Thus, swarm methods are more scalable and simple than auction mechanisms. Because of these advantages a lot of swarm like algorithms has been posed but, nowadays, those based on the so-called Response Threshold Method (RTM for short) are probably the most broadly used (see Section 2.2 for a detailed description of the classical swarm algorithm based on RTM). In these methods, each involved robot has an associated a task response threshold and a task stimuli.The task stimuli value changes over the time and indicates how much *atarctive* is the task for the robot. When the task stimuli, associated to a robot, takes a value greater than a certain threshold, the robot starts the execution of the task or, as happens in most cases, a robot selects a tasks with a probability functions that depends on the stimuli itself and the threshold [3]. To our best knowledge, in this system there are only two tasks and the robots can only choose between them. Hence, a robot can stay on its current task or change to the other one [16]. As can be seen, this decision making process can be dealt as a Markov chain with only two states. However, the probabilistic approach which yields support to Markov chains can become a handicap. In fact, if the robots has a number of tasks greater than two, such as happens in a real mission, then either the RTM based algorithms cannot be directly applied or to fit the transition probabilities in the Markov model can be a very hard labor and, hence, it is necessary to reason by means imprecise probabilities that are fixed subjectively. In addition to this handicap, the applicability of the Markov approach to real missions is also reduced because of transition probabilities must meet some typical constraints. One of them is that of the transition probabilities from one state to the other ones must be a probability distribution (the sum of all of them must be equal to 1) and, in general, this constraint is not satisfied in real problems. Despite, it is worthy to point out that in the literature can be found Markov chain decision processes in which normalized transition probabilities have been proposed in order to model the behaviour of multi-robot systems where there are more than two possible tasks (see, for instance, [13]). Although, this methodology implies to introduce unnatural manipulations of the original system. Furthermore, in real situations the transitions can be represented by numerical values outside the range of the unit interval. So the probability theoretical foundation may even be inappropriate.

In the light of the above-mentioned inexpedients associated to the probabilistic RTM for task allocation, it seems natural to search through literature a theoretical formalism, with a basis different from the probability theory, that may be useful to study in a natural way the task allocation problem via a RTM when more than two tasks are under consideration and, in addition, does not involve artificial requeriments or constraints as, for instance, the normalization. Fortunately, the desired formalism can be found in the literature and it is known as possibility theory (see [4] and [17]). For this reason this paper takes a first step towards a possibilistic theoretical formalism for a RTM and its utility for the MRTA problem. Concretely, the RTM will be implemented considering transitions possibilities instead of transitions probabilities and this fact will imply that in the intrinsic decision process the possibilistic Markov chains (also known as fuzzy Markov chains) play the role of the probabilistic ones. Moreover, a few powerful properties of this new method will be showed.

The remainder of the paper is organized as follows: Section 2 reviews the basics of the MTRA problem. Thus the relevant notation and the problem statement are introduced in Subsection 2.1. In Subsection 2.2 the swarm approach to the MTRA problem is presented. Concretely, the classical swarm algorithm based on the RTM is detailed and one limitation is discussed. Section 3 is devoted to developed the possibilistic theoretical formalism for a RTM and to show its utility for the MRTA problem. Specifically, in Subsection 3.1 formalizes a few concepts about possibilistic theory and fuzzy Markov chains that will play a crucial role in order to achieve our objective. Moreover, in Subsection 3.2, a specific MTRA problem is approached from a Swarm viewpoint via the use of fuzzy Markov chains and some relevant properties of such a method are studied. Besides, some typical cases of study are discussed in order to illustrate the utility of the new method. Finally, the conclusions and future work are presented in Section 4.

2 Multi-robot Task Allocation Problem

2.1 The Problem

In the literature there are a lot of MRTA problem definitions and all of them depending on the characteristics of the problem to solve (see [8]). One criteria, among others, to classify the MRTA problems consider the number of robots that can be assigned to each task [8]. Thus, if two or more robots can collaborate to carry out the same task, the problem is called "Multi-Robot Task" problem (MRT for short). Otherwise, if only one robot can be assigned to each task at the same time, the problem is called "Single-Robot Task" problem (SRT for short). This paper only considers, as a first approach to the possibilistic MRTA problem, the SRT problems that can be defined as follows.

Let $\mathbb{N}$ denote the set of positive integer numbers and let $n, m \in \mathbb{N}$. Denote by R the set of robots with $R = \{r_1, ..., r_n\}$ and by T the set of tasks to carry out with $T = \{t_1, ..., t_m\}$. A task allocation is a function $TA : T \to R$ such that $TA(t_i) \bigcap TA(t_j) = \emptyset$ provided that $i \neq j$. Observe that TA assigns to each

task $t_j \in T$ a robot $r_i \in R$ in such a way that no more than one robot can be assigned to the same task. Thus, the goal of a task allocation algorithm is to find an optimal task allocation TA^*, among all valid TA functions, which optimizes some system characteristics.

Following [6], the SRT problems can be described in terms of the well-known Optimal Assignment Problem (OAP). The OAP is defined in the following way:

Consider n robots (or agents in general) and m tasks to carry out, each robot can only be assigned to one task and each task requires only one robot. For each couple (robot-task), a value is defined that forecasts the robots performance for that task, that is, this value models the robots utility regarding that task. The goal is to assign a robot to each task to maximize the total utility U. This goal function is given by

$$U = \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq m} \alpha_{ij} U_{ij} w_j, \tag{1}$$

where $\alpha_{ij} = 1$ if the task i is assigned to agent j, and $\alpha_{ij} = 0$ otherwise; U_{ij} is the utility gained by the system when agent j is assigned to task i; and w_j is the weight or importance of the task j. Thus, w_j represents the priority of the task. The Hungarian's method [14] allows to get the optimal solution to this kind of problems in a time $O(nm^2)$ through a dynamic programing centralized method.

2.2 A Swarm Task Approach to MTA Problem: The Response Threshold Method

Although the Hungarian's method gives an optimal assignment which solves the OAP and, thus, the SRT problem, it is worthy to point out that it requires a central agent with global knowledge about the characteristics of all the robots and tasks to be performed. Moreover, since the environment where the robots are operating is dynamic (it can change over the time) the allocation algorithm should be executed constantly. Since in most of the real tasks it is needed a decentralized and very low computational cost task allocation algorithm, the preceding fact poses a handicap which is against the use of the Hungarian's method. In this direction methods based on swarm intelligence, as the RTM, are more useful and appropriate.

According to [1], the classical response threshold method defines for each robot r_i and for each task t_j, a stimuli $s_{r_i,t_j} \in \mathbb{R}$ that represents how suitable t_j is for r_i, where $\mathbb{R}$ stands for the set of real numbers. When s_{r_i,t_j} exceeds a given threshold ($\theta_{r_i} \in \mathbb{R}$), the robot r_i starts to execute the task t_j. To avoid relying on the threshold value to an excessive degree, the task selection is usually probabilistic. Thus, a robot r_i will select a task t_j to execute with a probability P_{r_i,t_j} which is given by

$$P_{r_i,t_j} = \frac{s^n_{r_i,t_j}}{s^n_{r_i,t_j} + \theta^n_{r_i}} \tag{2}$$

Figure 1 shows Equation (2) values as a function of s_{r_i,t_j} for several values of the exponent n and with $\theta_{r_i} = 50$. In order to reproduce the conditions used by most of the authors, n will be always equal to 2 (see [11]).

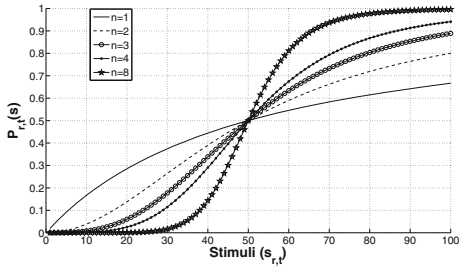


Fig. 1. P_{r_i,t_j} values (Equation 2) as a function of the stimuli for several n values with $\theta_{r_i} = 50$

.

As mentioned earlier, the classical response threshold method is only suitable when the robots have, at a given time, two available actions or tasks to perform, i.e., $T = \{t_1, t_2\}$. Therefore the probability P_{r_k,t_i,t_j} that the robot r_k ($k = 1 \ldots, n$) leaves the task t_i in order to perform the task t_j, with $i, j \in \{1, 2\}$ and $i \neq j$, can be calculated according to Equation (2) by

$P_{r_k,t_i,t_j} = \frac{s^2_{r_k,t_j}}{s^n_{r_k,t_j} + \theta^2_{r_k}}$ and $P_{r_k,t_i,t_i} = 1 - P_{r_k,t_i,t_j}$.

Of course, the evolution of the system over the time can be modeled by means of a finite Markov chain where, for each robot r_k, the transition matrix $P_{r_k} = \{Pr_k, ij\}^n_{i,j=1}$ is given by $Pr_k, ij = P_{r_k,t_i,t_j}$ for all $i, j = 1, 2$. Notice that, for each $k, i \in \{1, 2\}$, $P_{r_k,i1} + P_{r_k,i2} = 1$.

It is clear that the preceding approach fits perfectly to the case of two possible states of the system, i.e., $T = \{t_1, t_2\}$. However, when more tasks are available to be performed by the each robot over the time, i.e. $T = \{t_1, \ldots, t_m\}$ ($m > 2$) then it seems natural to ask whether in general, for each robot r_k and $i \in \{1, \ldots, n\}$, the equality $\sum_{j=1}^{m} P_{r_k,t_i,t_j} = 1$ holds. Obviously this constraint is violated in many real situations (see, for instance, [9]). In order to avoid this disadvantage, as we have mentioned in Section 1, normalization processes can be introduced although they imply to impose unnatural system modifications. Moreover, in addition to the aforesaid inconveniences, to determine the transitions probabilities is an arduous task in general and, therefore, reasoning with imprecise probabilities, which have to be fitted subjectively, becomes needful to model the behavior of the system. Finally, let us recall that when more than two tasks are under consideration then the system will evolve to a stable state provided that, for each robot r_k, the transition matrix P^{r_k} is regular (see [12]). In this case the behavior of the systems is described by the following result.

Theorem 1. *Let P be a transition matrix for a regular Markov chain. Then the matrices P^n approach a limiting matrix W with all rows the same vector w whose all components are strictly positive and their sum is equals 1. Moreover, $wP = w$.*

In the light of the preceding result, it must be pointed out that the evolution of the system to a stable state is only guaranteed asymptotically and, as a consequence, in many cases the knowledge of the final state of the systems is obtained by successive approximations. It follows that in the probabilistic framework the final state of the systems is known with some degree of approximation in many cases.

3 Possibility Theory and Task Allocation

Since the transition probabilities in multi-task allocation problems can violate in a natural way a few axioms of the probability theory and they may be also imprecise and fixed subjectively, in this section a RTM based on a more general framework than the probabilistic one is introduced. Concretely a possibilistic approach for task allocation is proposed in such a way that multi-task allocation problems can be formulated via RTM based techniques in which the above-mentioned handicaps disappear. The aforementioned framework allows to encode the imprecise nature of the transitions of the system and to model the transitions probabilities without incorporating unnatural manipulations in the spirit of the normalization process.

3.1 Possibility Theory and Markov Chains

Next we recall a few pertinent notions from the possibility theory which will be crucial to achieve our announced target.

On account of [4] (see also [17]), given a non-empty finite set Ω, a possibility distribution on Ω is a function $Pos : \Omega \to [0, 1]$ such that $\max_{\omega \in \Omega} Pos(\omega) \leq 1$. Moreover, provided that the power set of Ω is denoted by $\mathcal{P}(\Omega)$, a (non-normalized) fuzzy measure on Ω is a function $\mathcal{M} : \mathcal{P}(\Omega) \to [0, 1]$ which holds the following axioms:

i) $\mathcal{M}(\emptyset) = 0$;
ii) $\mathcal{M}(\Omega) \leq 1$;
iii) $\mathcal{M}(A) \leq \mathcal{M}(B)$ provided that $A \subseteq B$.

A fuzzy measure $\mathcal{M}$ on Ω is called a possibility measure whenever the additional axiom is satisfied:

iv) $\mathcal{M}(A \cup B) = \max\{\mathcal{M}(A), \mathcal{M}(B)\}$ for all $A, B \in \mathcal{P}(\Omega)$.

Of course, a possibility distribution Pos on Ω induces a non-normalized fuzzy measure on Ω denoted by $\mathcal{M}_{Pos} : \mathcal{P}(\Omega) \to [0, 1]$ and defined by

$$\mathcal{M}_{Pos}(A) = \max_{\omega \in A} Pos(\{\omega\})$$

for all $A \in \mathcal{P}(\Omega)$. Notice that $\mathcal{M}_{Pos}(\Omega) = \max_{\omega \in \Omega} Pos(\omega) \leq 1$.

Usually the probability of an "event" $A \in \Omega$ is understood as a degree of likelihood or frequency which A occurs. In contrast, the possibility of A, $\mathcal{M}_{Pos}(A)$, is related to our perception of the degree of feasibility of A occurs. Thus, $\mathcal{M}_{Pos}(A) = \max_{\omega \in \Omega} Pos(\{\omega\})$ and $\mathcal{M}_{Pos}(A) = 0$ mean that the event A is totally possible or plausible and impossible, respectively. Nevertheless, contrarily to the probabilistic approach, there may be two events A and B with $\mathcal{M}_{Pos}(A) = \mathcal{M}_{Pos}(B) = \max_{\omega \in \Omega} Pos(\{\omega\})$, i.e., both events are totally possible. Despite these notable differences between possibility and probability, there is a relationship between them. In particular, if an event is probable then it is certainly possible or, equivalently, the degree of possibility of an event is greater than or equal to its degree of probability.

The theory of possibility has been applied to model those decision making processes based on Markov chains recently in [2]. Next we recall the basic notions about possibilistic Markov chains (fuzzy Markov chains in [2]) because they will be crucial in our subsequent discussion.

Consider a system evolving in time in such a way that the states of the system is fixed and finite, say $S = \{s_1, \ldots, s_k\}$. Moreover, at any unit of time the system changes from one state to another one according the following memoryless possibilistic law:

If the system is in the state s_i at time n ($n \in \mathbb{N}$), then the system will move to the state s_j with possibility p_{ij} at time $n+1$. Moreover, the transition possibility p_{ij} does not depend upon which states the system was in before the current state s_i. Of course, this law yields that the future state of the system depends only on the present state. Furthermore, given that the system is in the state s_i at time n then the system will move to one of the $s_1, \ldots, s_k$ possible states at time $n+1$ and, hence, we have that $\bigvee_{j=1}^{k} p_{ij} \leq 1$ for every $1 \leq i \leq n$, where $\vee$ stands for the maximum operator on $[0, 1]$.

Note that the numerical value $\bigvee_{j=1}^{k} p_{ij}$ provides information about what is the most possible state at which the system will move from the state s_i. Although, of course, several states can become enjoy the same degree of possibility.

Next, let $x_i(n)$, $1 \leq i \leq k$, denote the possibility that the state s_i will occur at time n. Then it follows that $\bigvee_{i=1}^{k} x_i(n) \leq 1$ and, in addition, such a numerical value can be understood as the evolution of the system at time n is governed by the most possible state at that time. Notice that one of the states $s_1, \ldots, s_k$ must occur at time n. However, two states can become equally possible at time n and, thus, the dynamics of the system will be totally determined by the possibilities of transition. Obviously, the evolution of the system in time is given by the following equations

$$x_i(n) = \bigvee_{j=1}^{k} p_{ji} \wedge x_j(n-1) \tag{3}$$

for all $i = 1, \ldots, k$ and for all $n \in \mathbb{N}$, where $\wedge$ stands for the minimum operator on $[0, 1]$.

Obviously, the preceding systems of equations is equivalent to the below one in matrix notation:

$$x(n) = x(n-1) \circ P \tag{4}$$

for all $n \in \mathbb{N}$, where $P = \{p_{ij}\}_{i,j=1}^{n}$ is the matrix of transition possibilities, $\circ$ is the matrix product in the max-min algebra $([0,1], \vee, \wedge)$ and $x(n) = (x_1(n), \ldots, x_k(n))$ for all $n \in \mathbb{N}$. Observe that $x(n) = (x_1(n), \ldots, x_k(n))$ represents the possibility distribution of the set of states at time n (the ith component of $x(n)$ matches up with the possibility of the system is in state s_i at time n. Naturally, a possibility distribution $x(n)$ of the system states at time n is said to be stationary, or stable, whenever $x(n) = x(n) \circ P$.

The great advantage of the possibilistic Markov chains with respect to the probabilistic ones is given by the fact that under certain conditions the system which is modeled will converge to a stable (stationary) state in a finite time. In contrast probabilistic Markov chains under the hypothesis of regularity guaranteed that the system will converge to a stationary state asymptotically, i.e., not necessarily in a finite time (see Theorem 1). The conditions that provide the finite convergence character of the system in the possibilistic case can be found in [5]. With the aim of recalling such conditions let us introduce a few notions about fuzzy matrix.

Following [5], a matrix $A \in M_n([0,1])$ will said to be k-power-convergent if $A^k = A^{k+1}$ for some $k \in \mathbb{N}$, where A^k denotes the max-min composition of A and itself k times. Moreover, the least $k \in \mathbb{N}$ such that A is k-power-convergent (or simply power-convergent) will be denoted by $k(A)$ and called the index of A. Furthermore, given $A, B \in M_n([0,1])$, we denote by $A \leq B$ the fact that $a_{ij} \leq b_{ij}$, where $\leq$ stands for the usual order on $[0,1]$. Finally, a matrix $A \in M_n([0,1])$ will said to be column diagonally dominant provided that $a_{ii} \geq a_{ji}$ for all $i, j = 1, \ldots, n$.

Taking into account the above introduced concepts we are able to state the result which guarantees the aforesaid finite convergence character of possibilistic Marckov chains.

Theorem 2. *Let $A \in M_n([0,1])$. Assume that A is column diagonally dominant and that $A \leq A^2$. Then A is power-convergent and $k(A) \leq n-1$.*

In the light of the preceding result, those systems modeled by possibilistic Markov chains whose transition matrix satisfies the assumptions in the stament of Theorem 2 will evolve to a stationary state in a finite time.

It must be pointed out that the preceding result will be crucial in our subsequent discussion, since the equation that describes the evolution of a possibilistic Markov chain, Equation (4), is equivalent to the following one: $x(n) = x(0) \circ P^n$ for all $n \in \mathbb{N}$. So the study of the power-convergence of the transition matrix will play a central role in the study of the behavior of the system under consideration.

3.2 Possibilistic Markov Chains and Task Allocation

This section shows an example of how possibilistic Markov chains are useful for developing a RTM for solving multi-robot task allocation problems. For the

sake of simplicity, and honoring the the titile of the paper, a very simple but representative case is discussed. Finally, two illustrative examples, which worth to mention, are studied.

According to the MRTA problem statement consider a collection of robots $R = \{r_1, ..., r_n\}$ $(n \in \mathbb{N})$ and a set of tasks to carry out $T = \{t_1, ..., t_m\}$ $(m \in \mathbb{N})$. Assume that the tasks are randomly placed in an environment and the robots are initially randomly placed. Then the target is to find an optimal task allocation in such a way that only one robot can be assigned to each task at the same time. Hence our possibilistic RTM must decide which task must execute each robot although this decision must be made individually by each robot without exchanging information between them. Moreover, we will assume that each robot allocation only depends on the distance between the robot and the task.

From now on, denote by $x_i(n) = (x_{i1}(n), ..., x_{im}(n))$ a fuzzy set, where $x_{ij}(n)$ is the possibility for the robot r_i of executing the task t_j at time n. Consider the position space endowed with a metric d. Then denote by $d(r_i, t_j)$ the distance between the current position of r_i and the position of t_j and by $d(t_i, t_j)$ the distance between the position of the task t_i and t_j. Of course, it is assumed that when a robot is assigned to a task the position of this task and the robot's position are the same and therefore, the distance between the task and the robot is 0. Following the response-threshold notation, define the stimuli of the robot r_k to carry out task t_j as follows:

$$s_{r_k,t_j} = \begin{cases} \frac{1}{d(r_k,t_j)} & \text{if } d(r_k,t_j) \neq 0 \\ \frac{1}{\alpha_{r_k}} & \text{if } d(r_k,t_j) = 0, \end{cases} \tag{5}$$

where $\alpha_{r_k} = min_{t_j \in T_{r_k}} d(r_k, t_j)$ and $T_{r_k} = \{t_j \in T : d(r_k, t_j) \neq 0\}$. This stimuli s_{r_k,t_j} allows us to obtain, by means of Equation (2), the response threshold possibility

$$p_{r_k,ij} = \begin{cases} \frac{1}{1+d(r_k,t_j)^2\theta^2_{r_k}} & \text{if } d(r_k,t_i) = 0 \text{ and } d(t_i,t_j) \neq 0 \\ \frac{1}{1+\alpha^2_{r_k}\theta^2_{r_k}} & \text{if } d(r_k,t_i) = 0 \text{ and } d(t_i,t_j) = 0. \end{cases} \tag{6}$$

Notice that the numerical value $p_{r_k,ij} = \frac{1}{1+\alpha^2_{r_k}\theta^2_{r_k}}$ in Equation (5) can be understood in the following way: a robot r_k tends to stay on the task t_i when its position coincides with that of the task t_i. So the perception of the degree of feasibility of the robot stays at the position of the task t_i at the next time should be related to the greatest numerical value of the assigned possibilities. However, it is also possible that the robot r_k leaves the position of the task t_i, despite its current position coincides with that of the task t_i, and moves to the position of the task t_j to carry out it. This perception of the degree of feasibility that event occurs is represented by the numerical value $p_{r_k,ij} = \frac{1}{1+d(r_k,t_j)^2\theta^2_{r_k}}$. Note that when smaller the distance $d(t_i, t_j)$, with $i \neq j$ and $d(r_k, t_i) = 0$, the greater is the possibility degree of feasibility of the robot r_k transits from the position of t_i to the position of t_k.

From Equation (5) one can compute the fuzzy matrix of possibilities $P_{r_k} = \{p_{r_k,ij}\}_{i,j=1}^{m}$ which enjoys nice properties such as the next result shows.

Proposition 1. *Fix $r_k \in R$. Then P_{r_k} is power-convergent with $k(P_{r_k}) \leq m-1$.*

Proof. The fact that $0 \leq p_{r_k,ij} \leq 1$ for all $i, j = 1, \ldots, m$ immediately yields that $P_{r_k} \in M_m([0,1])$. Moreover, P_{r_k} is column diagonally dominant and satisfies that $P_{r_k} \leq P_{r_k}^2$, since $P_{r_k ij}^2 = \bigvee_{k=1}^{n}(p_{il} \wedge p_{lj})$ for all $l = 1, \ldots, m$. Therefore, by Theorem 2, we conclude that P_{r_k} is power-convergent with $k(P_{r_k}) \leq m-1$.

As a consequence of the preceding result we obtain the following conclusion.

Corollary 1. *Fix $r_k \in R$. Then the possibilistic Markov chain with transition matrix P_{r_k} converge to a stationary non-periodic solution in at most $m-1$ steps and, in addition, such a convergence does not depend on the initial posibilistic distribution $x_i(0)$.*

Finally, we focus our attention on two illustrative and interesting cases of study.

Case Study 1: Homogeneous robots and one robot per task. In this context homogeneous robots means that all the robots have the same threshold, that is, $\theta_{r_i} = \theta$ for $i = 1, \ldots, n$. Initially each robot r_k is assigned to a different task with a possibility $\frac{1}{1+\alpha_{r_k}^2 \theta^2}$. If we assume the same number of robots and tasks, $n = m$, then the robot r_k can initially be assigned to task t_k for all $k = 1, \ldots, n$. Moreover, it is clear that $p_{r_k,ij} = p_{r_l,ij}$ for all $k, l = 1, \ldots, n$. Furthemore, the initial possibility distribution of the states for robot r_k is given by

$$x_k(0) = (x_{k1}(0), \ldots, x_{kn}(0)),$$

where

$$x_{ki}(0) = \begin{cases} \frac{1}{1+\alpha_{r_k}^2 \theta^2} & \text{if } k = i \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

By Corollary 1 we have that the possibility distribution of sates for each robot will converge to a stationary non-periodic one in at most $n-1$ steps. In particular if case we take $n = 3$, then the possibility distribution of states for both robots will converge to a stationary non-periodic one in at most 2 steps. Specifically the possibility distribution of states at step 2 for each robot will be the following one:

$$\begin{aligned} x_1(2) &= (\tfrac{1}{1+\alpha_{r_1}^2 \theta^2}, \bigvee(p_{12}, p_{13} \wedge p_{32}), \bigvee(p_{13}, p_{12} \wedge p_{23})) \\ x_2(2) &= (\bigvee(p_{21}, p_{23} \wedge p_{31}), \tfrac{1}{1+\alpha_{r_2}^2 \theta^2}, \bigvee(p_{23}, p_{21} \wedge p_{13})) \\ x_3(2) &= (\bigvee(p_{31}, p_{32} \wedge p_{21}), \bigvee(p_{32}, p_{31} \wedge p_{12}), \tfrac{1}{1+\alpha_{r_3}^2 \theta^2})) \end{aligned}$$

Although all robots have the same transition matrix, all of them converge to a different possibilistic distribution of states. Despite in all cases the most possible

state is that of the robot stays performing the task where it was initially allocated, the robots have a non zero possibility for other tasks. Thus, all task has a possibility greater than 0 of execution.

Case of Study 2. Heterogeneous robots and all the robots are assigned to the same task. In this case all the robots has the same initial possibility distribution of states, for example all robots are initially allocated to task t_1. But, now the robots are heterogeneous, that is, $\theta_{r_i} \neq \theta_{r_j}$ provided $i \neq j$. Therefore each robot will have its own transition matrix P_{r_k}. Note that the number of robots and task may differ but if we take $n = m$, then, by Corollary 1, we have that the possibility distribution of sates for each robot will converge to a stationary non-periodic one in at most $n - 1$ steps.

In particular if we take $n = m = 3$, then the possibility distribution of states for all robots will converge to a stationary non-periodic one in at most 2 steps. Specifically if we assume, as an example, that all robots are allocated to task t_1, then the possibility distribution of states at step 2 for each robot will be the following one:

$$x_k(2) = (\frac{1}{1 + \alpha_{r_k}^2 \theta_{r_k}^2}, \bigvee(p_{r_k,12}, p_{r_k,13} \wedge p_{r_k,32}), \bigvee(p_{r_k,13}, p_{r_k,12} \wedge p_{r_k,23}))$$

for all $k = 1, \ldots, n$.

If the threshold had been the same for all the robots, all of them would converge to the same state, generating a non balanced system, that is, with a high possibility all the robots would have executed the same task (t_1). But now the robots are heterogeneous, so each vector $x_k(2)$ depends on θ_{r_k} and, therefore, the final distribution of the robots can be fitted through the threshold values.

4 Conclusions and Future Work

This paper has taken a first step to develop the theoretical basis for possibility multi-robot task allocation methods based on swarm intelligence. One of the most important swarm MRTA method is the response threshold. As mentioned in this paper, RT methods have a few inconveniences, from practical and theoretical point of view. In fact, if there are more than 2 states then easily either to determine the transitions probabilities is a hard task and, therefore, they have to be fitted subjectively, or the distribution of transitions is not probabilistic and must be normalized. This handicaps can be avoided using possibilistic Markov chains which clearly offers a more realistic and general approach because allow to model imprecise probabilities. The paper proves that, in the specific case where the transition possibility depends on a distance, the Markov process converges in a finite number of steps lower or equal to the tasks.

This work still has a lot of challenging aspects to add and to improve. For the time being, we focus on new scenarios and possibility transitions that depends on new factors like for example the utility of the task. We also plan to compare possibilistic Markov chains with current response threshold methods from an empirical point of view.

Acknowledgments. This work has been partially supported by projects DPI2011-27977-C03-03, TIN2013-42795-P and FEDER funding.

References

1. Agassounon, W., Martinoli, A.: Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In: 1st Int. Joint Conference on Autonomous Agents and Multi-Agents Systems, Bolonia, Italy, pp. 1090–1097, July 2002
2. Avrachenkov, K., Sanchez, E.: Fuzzy markov chainsand decision making. Fuzzy Optimization and Decisoin Maing **1**, 143–159 (2002)
3. Bonabeau, E., Sobkowski, A., Theraulaz, G., Deneubourg, J.-L.: Adaptive task allocation inspired by a model of division of labor in social insects. In: Lundh, D., Olsson, B., Narayanan, A. (eds.) Bio Computation and Emergent Computing, pp. 36–45. World Scientific (1997)
4. Dubois, H.P.D.: Fuzzy Sets and Systems: Theory and Applications. Academic Press (1980)
5. Duan, J.: The transitive clousure, convegence of powers and adjoint of generalized fuzzy matrices. Fuzzy Sets and Systems **145**, 301–311 (2004)
6. Gerkey, B.P.: On Multi-Robot Task Allocation. PhD thesis, Center of Robotics and Embedded Systems, University of Southern California, Los Ángeles, USA, August 2003
7. Gerkey, B.P., Mataric, M.: Sold!: Auction methods for multi-robot coordination. IEEE Transactions on robotics and Automation, Special Issue on Multi-robot Systems **18**(5), 758–768 (2002)
8. Gerkey, B.P., Mataric, M.: A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research **23**(9), 939–954 (2004)
9. Guerrero, J., Oliver, G.: Swarm-like methodologies for executing tasks with deadlines. Journal of Intelligent & Robotic Systems **68**(1), 3–19 (2012)
10. Heap, B., Pagnucco, M.: Repeated sequential single-cluster auctions with dynamic tasks for multi-robot task allocation with pickup and delivery. In: Klusch, M., Thimm, M., Paprzycki, M. (eds.) MATES 2013. LNCS, vol. 8076, pp. 87–100. Springer, Heidelberg (2013)
11. Kalra, N., Martinoli, A.: A comparative study of market-based and threshold-based task allocation. In: 8th International Symposium on Distributed Autonomous Robotic Systems, pp. 91–102, Minneapolis, USA, July 2006
12. Kemeny, J., Snell, J.: Finite Markov Chains. Springer-Verlag (1960)
13. Kim, M., Baik, H., Lee, S.: Response threshold model based uav search planning and task allocation. Journal of Intellent Robotics Systems **75**, 625–640 (2014)
14. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1), 83–97 (1955)
15. Navarro, I., Matía, F.: An introduction to swarm robotics. ISRN Robotics, 2013 (2013)
16. Yang, Y., Zhou, C., Tin, Y.: Swarm robots task allocation based on response threshold model. In: International Conference on Autonomous Robots and Agents, pp. 171–176, Willengton, New Zeland, February 2009
17. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems **1**, 3–28 (1978)

A Bottom-up Robot Architecture Based on Learnt Behaviors Driven Design

Ignacio Herrero(✉), Cristina Urdiales García,
José Manuel Peula Palacios, and Francisco Sandoval Hernández

Dpt. Tecnología Electrónica, University of Málaga, Málaga, Spain
iherrero@uma.es
http://www.grupoisis.uma.es

Abstract. In reactive layers of robotic architectures, behaviors should learn their operation from experience, following the trends of modern intelligence theories. A Case Based Reasoning (CBR) reactive layer could achieve this goal but, as complexity of behaviors increases, the curse of dimensionality arises:too many cases in the behaviors casebases degrade response times so robot's reactiveness is finally too slow for a good performance. In this work we analyze this problem and propose some improvements in the traditional CBR structure and retrieval phase, at reactive level, to reduce the impact of scalability problems when facing complex behaviors design.

Keywords: Case based reasoning · Reactive layer · Learning architecture · Robotics

1 Introduction

Hybrid architectures are currently the most successful approach to control systems in modern robots. Most hybrid systems have a planner layer, related to abstract knowledge and global objectives of the robot in the long-term, and a reactive layer, which decides the response of the robot at local short-term scale, and is composed by primitive behaviors. However, behaviors at the reactive layer are usually designed at earlier stages of the architecture development in a fixed algorithmic way, with little or no adaptability to changes in environment and problem evolution. There are some evidences which suggests that these reactive components should be able to adapt by learning from the robot experiences. First, Ethology shows us that some animals behaviors are innate-with-memory behaviors which must be tuned by learning or experience: this is how a baby bee learns its hive look and how to navigate to and from it [10]. Furthermore, modern theories about human intelligence claim that even in a complex deliberative response, just a few neurons are involved, with a not very fast transmission of information. So human brain's calculation power is, a priori, orders of magnitude lower than modern computers'. Instead, human intelligence appears to be funded on *learning from experience*, by storing everyday events, knowledge,

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 159–170, 2015.
DOI: 10.1007/978-3-319-19258-1_14

and responses in a powerful data base imprinted on our brain, and retrieving them when in a similar situation [5]. This view of human intelligence has been assimilated to research in Artificial Intelligence (AI), especially in the field of Intelligent Robot Systems (IRS).

There are many approaches to robotic hybrid architectures which follow these learning guidelines [14][11], but most of them focus this learning on deliberative levels. Human brain follows a "nest structure", so its different areas should operate with the same principles no matter what are devoted to; they only differ in the nature of the information and the abstraction level of the concepts they manage, which defines their situation on the reasoning chain [5]. It seems reasonable to think about an IRS architecture in which all the layers and sub-layers at every level,*including the low reactive level*, are based on some kind of learning by experience. Besides the improvement in flexibility and adaptability at this reactive level, the architecture would gain in homogeneity and scalability, specially if the same AI method is used for the layers development. In [12] we proposed a first approach to a CBR-learning based implementation of the reactive layers for a hybrid architecture. Despite its good performance, it showed some problems related to the ad-hoc nature of the behaviors design, such as scalability problems with the increase of information and retrieval time of the response. In this work, we suggest some improvements to solve these problems, mainly related to changes in the casebase structure and retrieval process, but also in switching from numeric to conceptual knowledge in order to extend the model, in the future, to higher deliberative layers.

2 CBR Based Reactive Layer

Reactive layers rely on coupling sensors information and actuators response into a set of low level primitive modules or *behaviors*. Every behavior deals only with local, short-term information related to the goals or scope of the module [9]. Reactive Behaviors are very well suited to fast low-level decisions, so this layer is specially important in dynamic unstructured environments. These low level behaviors are also combined in a bottom-up way to produce more complex emergent ones. There are different methods to achieve this emergence, from combining or switching several behaviors [6], to using subsumption architectures [3]. But these methods usually depend on many parameters that need to be optimized for each specific problem as they depend on the robot kinematics and dynamics, sensor calibration, and even mechanical errors. For these reasons many reactive architectures and layers have included AI tools such as fuzzy-logic rules [2] or ANNs [8] to fine-tuning the proposed behaviors through experience. Fuzzy-logic methods define a set of rules to map sensor reading and motor actions, rules whose parameters can be adjusted during the robot execution. But it's difficult to accomplish a correct design of such rules, as they need a great knowledge about both fuzzy-logic and the problem domain. ANNs get over these obstacles as, once the structure of layers and neurons of the network has been chosen, the training process is transparent to the user. However, ANNs only provide us of

a better or worse operating model, but are black boxes with no clues of what has gone wrong when they don't find a right solution [5]. On the other hand, Case Based Reasoning (CBR) combined with learning, could allow us to obtain a fully-working system, as well as to understand the learning process and the knowledge acquisition, which could very useful to debug errors and malfunction of our system.

2.1 Experience Based Learning: CBR

Case-Based Reasoning is a reasoning, learning and adaptation technique to solve current problems by retrieving and adapting past experiences [1]. A new problem together with its solution can also be stored in the casebase,as a new case, to be used later. Thus, better solutions can be derived when faced against less experienced situations. As we can see, the philosophy of CBR is quite in tune with the model of intelligence as prediction, and the learning method to build it. A CBR system cycle to solve a new problem consists of four steps: (i) retrieve the most similar stored case to the new current case; (ii) adapt its solution to the new current case; (iii) evaluate the results of the proposed solution; (iv) learn from the new experience. Consequently, when creating a new CBR application, design decisions often concern: i) how to describe the problem to solve (ii) which is the casebase or case library structure; within a particular case structure; (iii) how retrieval process and similarity assessment between cases can be evaluated; (iv) how to adapt the old solution to solve the new current problem, (v) how to evaluate the success of the proposed solution, and (vi) what to learn and how to learn from solved problems.

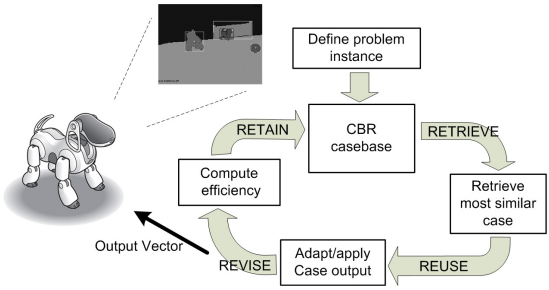


Fig. 1. CBR cycle

CBR has been widely used in many experience learning frameworks in Robotics [7][13]. In [12], we proposed a reactive CBR-learning layer in which coupling of visual data and motion commands were learned by low level reactive behaviors. This way, the robot could know what to do from the perception it had at any moment. Such learning was done by supervised training, but from the own robot experience too, so not only the goal of the behavior is learned, but

also the environmental and the own robot physical conditions which affect the attainment of such goal. Although our method showed a good performance, its advantages were not evident because it was tested with fairly simple behaviors that could be implemented analytically. In our new proposal we intend to systematize the design of different behaviors when several visual elements (goals, rival and friend players,ball, field,..) are included in the CBR casebase which controls the behavior operation. This means that CBR must learn how every individual object influences the operation of the behavior, but also how *their relationship* affects it, too. As there are more components in a CBR case, the number of required cases increases from nearly a hundred to thousands: scalability problems arise, as CBR cannot respond as quick as a reactive layer would need. Casebase size can be reduced by clustering or discretization which also implies a transition from a numerical representation of the information to a conceptual one. This way, concepts at low levels would combine to develop more abstract complex ones at higher level behaviors, which would emerge from the lower layer in a bottom-up integration. We propose some changes in the original structure and operation of the CBR casebase in order to further speed the process of recovering the solution at the module when a new problem is presented: i) from a flat database structure we have switched to a hybrid hierarchical-flat solution, using the concept of "context" to traverse the tree hierarchy; ii) we have also split the CBR recovering phase in two sub-phases: one fast scene recognition by indexing objects; and the traditional search of the most similar case by a distance comparison.

3 CBR Behavior Design

The scenario for our experiments is similar to the one described in [12], in which an AIBO ERS7 robot is also used. In this work, mainly visual information will be considered, but we will add a IR chest sensor too, to detect when the robot is in possession of the ball. The experiments will be run in a Robosoccer field in which, besides from the ball and two goals, a maximum of two opponent robots and a friend robot, could appear. The inclusion of all or only some of these elements in the CBR cases depends on the behavior definition and goals.

3.1 Case Definition

Case definition requires an input instance, related to the necessary knowledge for the behavior operation; a solution which defines the response of the behavior, either a concept or a motor action; and some measure of efficiency which is also related with the performance of the behavior in the fulfillment of its goal.

In our vision-based robot the input cannot be the whole image frame or a sub-sampled version, as its dimension would be proportional to pixel resolution. Using Principal Component Analysis (PCA), histograms, or color moments would help, but redundancy could affect negatively the performance of behaviors, as they are supposed to work only with local information. Qualitative techniques

have been proposed as a better option for active vision applications, but input feature selection in qualitative methods depends heavily on the specific domain of the goal or problem to solve. Selected features should be representative, easy to obtain, and robust against artifacts such as changes in light conditions or occlusions. In our example, possible input features include descriptions of the objects found at each image. To obtain that description, images are codified in a HSV space and a fast segmentation algorithm is run to obtain different color blobs, represented by their centroid and area, plus possible clipping in case the object is at the boundary of the image; data is normalized to their extreme values. The case includes also the normalized PAN and NOD angle of the robot head, as it affects the relative positions of objects in a frame (see figure 2).

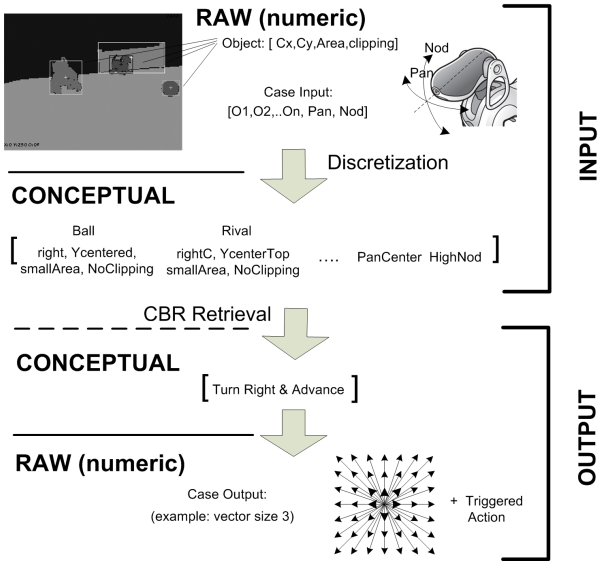


Fig. 2. Case definition: Input instance elements and output response

To reduce the amount of cases, input instance components are discretized/ indexed; this process allows us to turn knowledge from numeric data to conceptual information. The area of an object, for example, is not represented by its normalized number of pixels, but by a description such as "minimum area" or "medium area" which, besides of making easier to debug the "reasoning" process of the robot, allows to develop more complex concepts to describe the state of the environment and robot with respect to the goal of the behavior, i.e., "minimum area" and "high Y-centroid" would represent a "far object". Discretization is a key element for the definition of the input instance, and is also dependent of the goals and design of the behavior. At the moment a human expert heuristically decides the number of indexes and their boundaries for every component in the considered behavior. In the future would be desirable that such decisions were

automatically taken by means of statistical analysis of the continuous input data obtained from the supervised learning stage.

In our original proposal the output of a case was a single vector of instant motion, expressed by components of rotation, translation and strafe, which defined the robot trajectory until a new CBR consultation was done. This single vector output has a lot of inconveniences regarding a possible combination of several behaviors output in a similar way to a Potential Fields approach (PFAs), as is very prone to result in a null motion in legged robots. For this reason we use now a *chain of vectors* (figure 2), to represent a more complex movement pattern, but also easier to combine with similar patterns from other low level behaviors. Size of the chain would depend on the desired granularity of the output but also on the reactiveness of the robot response. With a too long chain, CBR consultations would be too sparse in time to obtain a fast enough reactive overall behavior. Output also includes additional values to represent special actions (shoot, dive,..) triggered in specific situations. Finally, an efficiency function at the behavior module weights the utility of the output of a case, related to the goal of the behavior. Specific components of efficiency functions are behavior dependent, so they are usually defined by human experts, considering not only the objects and elements linked to the behavior, but also concepts as "softness" or "security" in the robot motion. Aspects related to efficiency function and case adaption will be addressed in future works.

3.2 Casebase Structure and Case Learning

Casebase structure defines the organization of the cases at the database, and plays an important role in the retrieval phase of the CBR cycle. Stored cases are usually organized either on a flat memory or a hierarchical one. In a flat memory cases are stored in a simple list, and the retrieval is done by comparing the problem with every case stored at the casebase, and selecting the case best matching the problem description. This comparison is done using a distance or similarity function over the components of both the problem and each stored case. Despite the simplicity of this approach and the easiness of addition of new cases, there are scalability problems when the number of stored cases grow up, as the recovery time is proportional to such amount. In hierarchical memories, cases are stored in the form of a graph or tree, in which every branch and leaf corresponds to a component of the case. Hierarchical memories provide a more efficient case retrieval, but this process could miss some adequate cases while searching, because once you have choose a path, the cases you can access to are a subgroup of the total, thus making impossible to retrieve a case included in other branch of the tree.

While in [12] we used a flat structure for the casebase of our behaviors, the increase in complexity of the cases at the new proposed behaviors, drove us to find another type of structure which allows to manage a great number of cases without being detrimental to the speed of retrieval phase. Instead, we have considered a compounded structure that mixes both viewpoints: our database is distributed over a tree in which final leafs are flat sub-casebases (see figure 3).

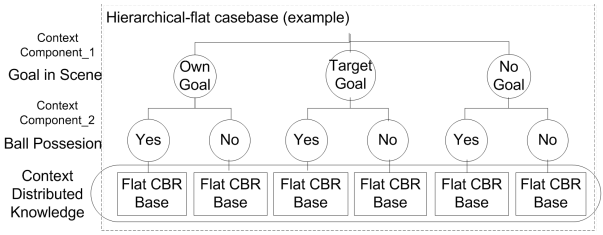
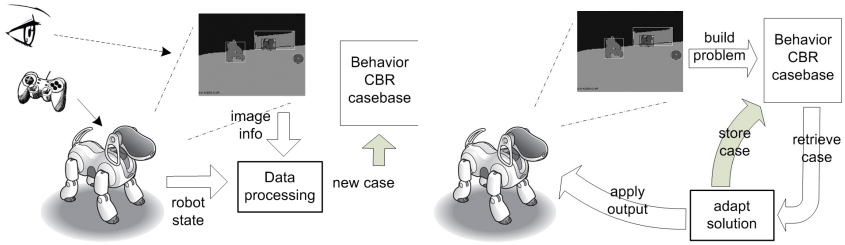


Fig. 3. Mixed Hierarchical-Flat casebase structure

Traversing trough the tree is done by means of specific context information, which can be obtained from the problem presented to the behavior. This information is mutually exclusive, so there is no possibility of ending in a wrong leaf-casebase. Information context could be, for example, the presence of the own goal or the rival goal, which would be related to a very different set of cases, which would be separated in different leaf casebases. As we can access the right casebase very fast, we achieve a great improvement in casebase organization and retrieval with a minimum cost. Nevertheless, we must point out that is not always possible to find context components which allow to build the tree part of this mixed structure, so not all behaviors could enjoy this benefit.

Knowledge at the CBR casebase is obtained in a two stage procedure (figure 4). First, a *learning by observation* approach is taken to seed the casebase with a set of initial cases. In this training phase, a trainer guides the robot using a conventional joystick to operate the desired behavior. The casebase is seeded from the cases learned during the different runs. The main advantage of this learning approach is that humans implicitly cope with kinematics and dynamics related to the robot, which are not easy to parameterize. So the robot, through this learning, can absorb them as well. Also, the intrinsic mechanisms related to the behavior are automatically incorporated to the robot knowledge without fully understand their rules and parameters. We can also manually add cases to the casebase, but a good understanding of the case components influence over the behavior operation is needed to do this. When a primary casebase has been obtained and the robot can run a basic operation of the behavior, a second stage of learning-by-own-experience is performed. The robot works in autonomous mode with no external supervision, retrieving from the behavior's casebase the most similar case to the current situation of the environment and robot state. If the retrieved case is not enough similar to the problem, it means that the robot is facing a new situation, so the output of the retrieved case should not be directly applied, but *adapted* instead. When adapted, output is applied and its performance is evaluated using a *efficiency function* which decides if the new case is good enough to be incorporated to the casebase.



(a) Learning by Observation (b) Learning by own-experience

Fig. 4. Two stage learning in our proposal framework

3.3 Retrieval Algorithm and Similarity Functions

A key aspect for the success of a CBR reactive system depends on having a good retrieval algorithm which can recover the more suitable case, but also fast enough to allow a good response time. Flat casebases have a scalability problem because the problem must be compared to every case stored at the casebase so, when its size grows up too much, the response time can be too high for a good reactiveness of the module. Although discretization and/or clustering of the input instances can relieve this problem, complex behaviors could still end with a high number of stored cases. The proposed mixed tree-flat structure of the casebase improves the retrieval time, as total knowledge is distributed among the leaf casebases, reducing the number of cases to compare in the retrieval stage. Once in the right leaf-casebase, and to further improve the retrieval operation at the CBR casebase we have introduced a two stage similarity function. Traditional similarity functions quantify the similarity of two cases by computing the distance between each of its components in a multidimensional space, weighting every local distance according to the influence of the component on the case representation, and aggregating weighted component distances using different functions. The choice of the distance function is problem dependent, and there is not a clear translation from one domain to other. An influential hypothesis has been that Euclidean distance is valid when stimulus dimensions are perceptually integral (such as the brightness and saturation of a color), whereas city-block distance is appropriate when stimulus dimensions are perceptually separable (such as the color and shape of an object) [4]. In our proposal we use a Manhattan normalized distance to compare problem and case similarity, but previously we do a pre-filtering stage in order to reduce the possible candidates to the recovered solution by means of a feature based distance, a *retrieval by objects in scene*. In this first stage, objects in scene are identified and a binary index with their presence (1) or not (0) is build. The descriptors of the elements related to the behavior are placed always in the same position, which is also used to represent the presence of the object in a string. Every case in the casebase has an associated index string representing the objects in the associated scene. When a problem is presented, the corresponding index string is generated and compared

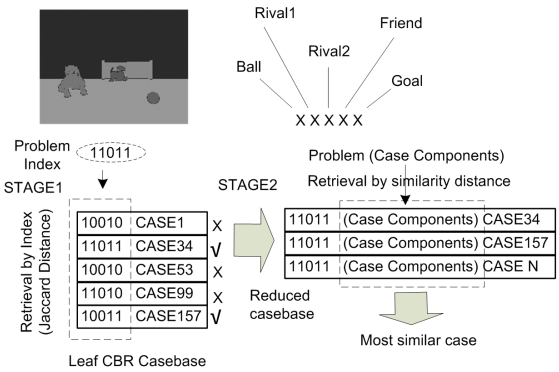


Fig. 5. Two stage retrieval operation at the CBR casebase

to the cases' ones using a Jaccard distance. A list with the most similar cases is generated, so the second retrieving stage carries out only on a subset of the casebase contents, thus improving the speed of the overall retrieving operation (figure 5).

4 Experiments and Results

In this section we present an example of behavior design following the guidelines given above. The behavior is defined in the controlled framework of a Robosoccer application in which two goals, a ball, maximum of two rival players and one team player are considered. The goal of the proposed behavior will be reach the ball with the presence of other players in the field. This behavior is quite more complex than the original one of just reaching the ball alone, as it is not enough to aim the ball and walk straight to it; depending of the rivals and friend position more complex maneuvers must be achieved in order to, not only reach the ball, but circle an obstacle, block the rival and/or face open field when the ball is finally captured. The input instance is defined by the presence of ball and player objects; goals won't be included as we consider they don't affect the performance of this specific behavior. The case input includes also pan and nod of the robot head, as stated in section 3.1. As behavior is driven by visual information, a tracking function has been developed to automatically move the head in order to maintain on-board camera focused on objects which are significant to the behavior. Case output will be a chain of three temporally consecutive motion vectors. Learn by observation is achieved through several runs from different starting positions of our robot, ball and obstacles. In figure 6 we can see a set of diverse scenarios trained for the considered behavior, together with an example of a run experiment.

Cases are acquired at a rate of 4 per second, and intervals of discretization were chosen analyzing the evolution of the continuous data taken at the training phase. Initially we considered five equally spaced intervals or classes at each case

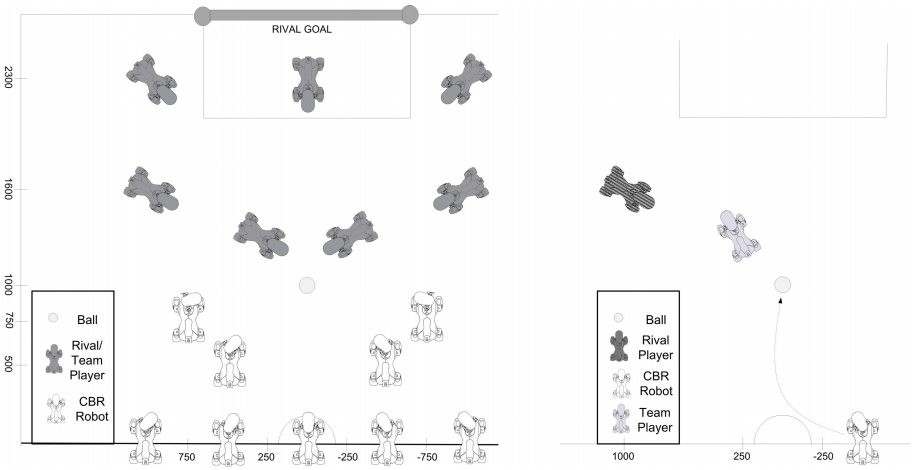


Fig. 6. Set of trained situations. Example of a learned run

component; but after an analysis of some runs one more class was added to the Pan component to break symmetry in its description; and area of objects was finally categorized into seven classes with interval size increasing in a quadratic form. Categorization allowed to compress knowledge from about twenty five thousand cases to barely three thousands. Categorized cases with the same input and different output were grouped by a majority vote.

Robot performance in autonomous mode was almost perfect from the same trained situations, but also in a great variety of similar ones . Nevertheless, a few scenarios that included input instances too different to the stored at the casebase showed a poor attainment, proving the necessity of a second stage of learn-by-own-experience case acquisition, as a complete supervised training at all possible situations is neither possible nor desirable. Also, categorization of the database allowed us to debug the execution of the experiments in a easy way.

In figure 7 we can see the improvement in retrieval time using our casebase structure and modified CBR retrieval method, with respect to the original flat casebase, under different situations. We gradually increased the complexity of the learned behavior by adding more objects and situations to the scenario, thus increasing the amount of knowledge needed at the behavior. We can see the evolution when space dimension of discretized case components grows, which increase also the dimensionality of possible cases at the behavior's casebase. We tested two discretization levels for the case components: a coarse one, as stated in the previous paragraph; and a fine one, in which we doubled the categories or discretization levels of each component, with respect to the coarse one. Retrieval time at the flat casebase is $O(cn)$, where c is the number of components of a case, and n the amount of cases at the casebase. Using our two stage procedure we can reduce the set of candidates to the second stage, thus improving the final retrieval time. The enhancement is proportional to the amount of cases in the

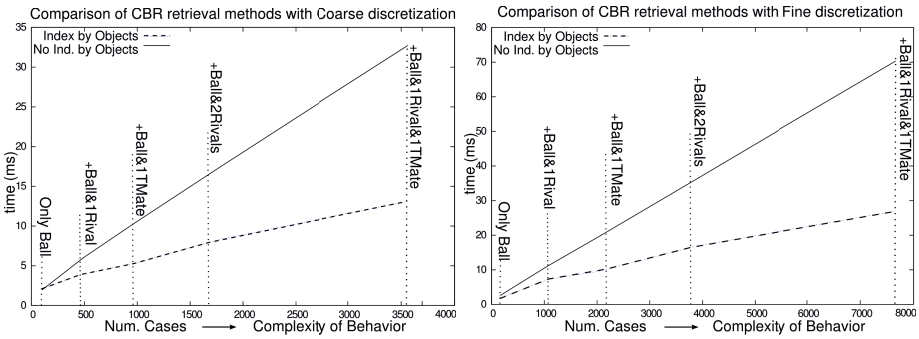


Fig. 7. Comparison of retrieval times using a two stage retrieval method

sub-sets after stage one, and only situations with all cases in just one sub-set would show a worse performance. Increasing the complexity of the behavior, with more objects and case components, and/or a finer-grained discretization, would magnify the advantages of our approach, as dimensionality of the space of possible cases would be greater.

5 Conclusions and Future Work

In this work we have presented an approach to design CBR-learning reactive behaviors for vision-based autonomous robots, when complexity of the target behavior increases, as there are more elements involved in the behavior's design, and scalability problems arise. This complexity affects the dimensionality of the possible cases which hold the knowledge of the behavior, and increases retrieval time which is directly related to this aspect. A good reactive behavior should respond quick enough to changes in environment and problem conditions, so a too-long retrieval time is unacceptable in any reactive layer. In order to solve these problems we have proposed a qualitative representation of the case components, by means of a indexation/discretization process which allow us to reduce the maximum amount of possible cases at the CBR casebase, and also describe the behavior's scenario using abstract concepts instead of numbers. This description, besides making easier the design and debug process of the behavior, would allow us to build more and more complex concepts by aggregating simpler ones, in a "nest-structure" fashion, which agrees with modern theories or human intelligence structure and organization. We have verified the good performance of our previous work, under more tight conditions, with the development of an example of a complex behavior. This complex behavior allowed us to show how scalability problems could arise and compromise the good performance of the behavior reactiveness. We have proposed a new mixed hierarchical-flat casebase structure and a two stage case retrieval algorithm, in order to speed up the CBR operation and allow a good behavior response, despite the increasing of information inside the casebase. We have proved the validity of this scheme, whose benefits

become more evident when the size of the casebase increases, as we have shown in different experiments.

Future work will focus on combination of different reactive behaviors to obtain an emergent complex one, which would both represent a higher level concept build from its compounding behaviors, but also an instruction of the actions the robot should take at this emergent behavior, derived from the corresponding actions at the low-level behaviors. Furthermore, we want to study more exhaustively the learning-by-experience stage, to improve new knowledge acquisition and casebase maintenance at the behaviors. In order to do so, we must develop better case adaption algorithms and adequate efficiency functions for the new adapted cases.

Acknowledgments. This work has been partially supported by the Spanish Ministerio de Educacion y Ciencia (MEC), Project n.TEC2011-29106 and Andalucia TECH:Campus of International Excellence.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications **7**(1), 39–52 (1994)
2. Aguirre, E., González, A.: Fuzzy behaviors for mobile robot navigation: design, coordination and fusion. Int. J. Approx. Reason. **25**(3), 255–289 (2000)
3. Brooks, R.: A robust layered control system for a mobile robot. IEEE J. Robot Automat. **2**(1), 14–23 (1986)
4. Garner, W.R.: The processing of information and structure. Halsted Press, The Experimental Psychology Series. L. Erlbaum Ass. (1974)
5. Hawkins, J., Blakeslee, S.: On Intelligence. Owl Books (2004)
6. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: IEEE 1985 Int. Conf. Robot, vol. 2, pp. 500–505 (1985)
7. Kruusmaa, M.: Global navigation in dynamic environments using case-based reasoning. Autonomous Robots **14**(1), 71–91 (2003)
8. Low, K.H., Leow, W.K., Ang Jr., M.H.: A hybrid mobile robot architecture with integrated planning and control. In: Int. J. Conf. Auton. Agent Multi. Ag. (AAMAS 2002), pp. 219–226, NY, USA (2002)
9. Mataric, M.J.: Interaction and Intelligent Behavior. PhD thesis, Department of Electronic Engineering and Computer Sciencie (1994)
10. Murphy, R.: Introduction to AI Robotics. MIT Press, Cambridge (2000)
11. Murray, J.C., Erwin, H.R.: Wermter., S.: Robotic sound-source localization architecture usingcross-correlation and recurrent neural networks. Neural Networks **22**(2), 173–189 (2009)
12. Peula, J.M., Urdiales, C., Herrero, I., Sánchez-Tato, I., Sandoval, F.: Pure reactive behavior learning using case based reasoning for a vision based 4-legged robot. Robot. Auton. Syst. **57**(6–7), 688–699 (2009)
13. Ros, Raquel, López de Màntaras, Ramon, Arcos, Josep-Lluís, Veloso, Manuela M.: Team Playing Behavior in Robot Soccer: A Case-Based Reasoning Approach. In: Weber, Rosina O., Richter, Michael M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 46–60. Springer, Heidelberg (2007)
14. Wang, M., Liu, J.N.K.: Fuzzy logic-based real-time robot navigation in unknown environment with dead ends. Robot. Auton. Syst. **56**(7), 625–643 (2008)

From Human Eye Fixation to Human-like Autonomous Artificial Vision

Viachaslau Kachurka[1,2(✉)], Kurosh Madani[1], Cristophe Sabourin[1], and Vladimir Golovko[2]

[1] LISSI / EA 3956 Laboratory, Senart-FB Institute of Technology, University Paris-Est Creteil, Lieusaint, France
viachaslau.kachurka@univ-paris-est.fr, {madani,sabourin}@u-pec.fr
[2] Neural Networks Laboratory, Intelligent Information Technologies Department, Brest State Technical University, Moskowskaya st. 267, 224017 Brest, Belarus
{vakochurko,gva}@bstu.by

Abstract. Fitting the skills of the natural vision is an appealing perspective for artificial vision systems, especially in robotics applications where visual perception of the surrounding environment is a key requirement. Focusing on the visual attention dilemma for autonomous visual perception, in this work we propose a model for artificial visual attention combining a statistical foundation of visual saliency and a genetic optimization. The computational issue of our model relies on center-surround statistical features calculations and a nonlinear fusion of different resulting maps. Statistical foundation and bottom-up nature of the proposed model provide as well the advantage to make it usable without needing prior information as a comprehensive solid theoretical basement. The eye-fixation paradigm has been considered as evaluation benchmark providing MIT1003 and Toronto image datasets for experimental validation. The reported experimental results show scores challenging currently best algorithms used in the aforementioned field with faster execution speed of our approach.

Keywords: Autonomous vision · Center-surround saliency · Evolutionary optimization · Eye fixation · Human-like visual attention

1 Introduction

An appealing area of research is the human attention modelling (and especially, human visual attention modelling), which has benefited from a substantial gain of interest over last decade because of its potential for various domains of applications. The related wealthy scientific literature gathers numerous works highlighting an intense research activity in this field. In their work [1] Borji and Itti made an overview of the human attention modelling problem and established a classification of subtasks in this field, differentiating more than 60 different models into 28 subsets of methods. Some of these subsets depend (at least partially) on detection of salient objects in scene [15,16]. Other subsets deal with

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 171–184, 2015.
DOI: 10.1007/978-3-319-19258-1_15

detecting (or defining) the most interesting spot in scene, which may or may not be a part of the most salient object of this scene [11,13].

Difference between the tasks highlights the vast variety of mechanics, used in models — starting from bottom-up versus top-down driven models [6,25], ending with diverse behavioural and parametric (non-behavioural) techniques used in models. A number of listed models covers different subtasks. For example, the approach described in [28] covers both object detection problem and eye fixation prediction problems.

In the same way, our previous work relating the fire detection [12], or more generally the detection of complex, deformed and multi-shaped objects, may cover "scene classification", "salient object detection" or "interesting point detection" accordingly to overview presented by [1], while according to [27], not being any of the above-mentioned tasks, but representing somehow some similarity with the task of eye-fixation prediction due to high overall saliency of the fire which, with high probability, will be an eye-fixation point in images containing fire.

Focusing on the dilemma of visual attention for autonomous visual perception, in this work we propose a model for artificial visual attention combining a statistical foundation of visual saliency and a genetic optimization. Statistical foundation and bottom-up nature of the proposed model provide the advantage to make it usable without prior information, making it suitable for usage in autonomous artificial vision in general. Although focusing distinct purpose and different applications, the "eye-fixation" paradigm has been considered as evaluation benchmark for experimental validation. In fact, considering that eye-fixation will follow what may be considered as relevant in visually perceived information, this paradigm provides an appealing experimental credentials linking "human-like" visual skills. We provide experimental results on MIT1003 [9] and Toronto [3] image datasets and a comparison to currently best algorithms used in the aforementioned field. The reported results show evaluation scores comparable to the state-of-art algorithms with advantage of a comprehensive solid theoretical basement and fast execution speed of our approach.

In Sect. 2 we introduce the groundwork of the proposed approach. The section will first expound a statistical foundation of visual saliency, then a dual analysis of analogy between "salient object detection" and "eye-fixation" tasks will be presented. Finally, the proposed approach will be described as ending this key-section of the paper. The Sect. 3 will focus the obtained results. Finally, Sect. 4 will conclude the paper sketching further perspectives of the presented work.

2 Theoretical Basis of Statistically Driven Artificial Visual Attention and the Proposed Approach

The proposed approach and related algorithm is based on the statistic-driven center-surround concept for objects detection, defined in [18] and largely inspired from introduced notions. Although successfully applied to different contexts [12, 18,19], raw algorithm proposed in [18] could not handle the purpose of present paper.

While the segmentation and object extraction tasks are parts of the related algorithm, we will not discuss these two operations in frame of this paper (details can be found in [18,19]).

2.1 Saliency Map Computation

Let us suppose the image Ω_{YCC}, represented by its pixels $\Omega_{\mathrm{YCC}}(x)$ in YCrCb color space, where $x \in \Psi^2$ denotes 2D-pixel position. Let $\Omega_{\mathrm{Y}}(x)$, $\Omega_{\mathrm{Cr}}(x)$ and $\Omega_{\mathrm{Cb}}(x)$ be the colors values in channels Y, Cr and Cb, respectively. Similarly, we can define $\Omega_{\mathrm{RGB}}(x)$, $\Omega_{\mathrm{R}}(x)$, $\Omega_{\mathrm{G}}(x)$ and $\Omega_{\mathrm{B}}(x)$ in RGB color space. Finally, let $\overline{\Omega}_{\mathrm{Y}}$, $\overline{\Omega}_{\mathrm{Cr}}$ and $\overline{\Omega}_{\mathrm{Cb}}$ be median values for each channel throughout the whole image.

We can calculate two different features, Global saliency map, and Local saliency map, which represent different techniques - overall contrast and local center-surround contrast of each region of image.

Global Saliency Map (GSM). GSM, denoted $M_{\mathrm{G}}(x)$, is a result of non-linear fusion of two elementary maps $M_{\mathrm{Y}}(x)$ and $M_{\mathrm{CrCb}}(x)$, relating luminance and chromaticity separately. Equations (1) – (3) detail the calculation of each elementary map as well as the resulting GSM.

$$M_{\mathrm{Y}}(x) = \|\overline{\Omega}_{\mathrm{Y}} - \Omega_{\mathrm{Y}}(x)\| \tag{1}$$

$$M_{\mathrm{CrCb}}(x) = \sqrt{\left(\overline{\Omega}_{\mathrm{Cr}} - \Omega_{\mathrm{Cr}}(x)\right)^2 + \left(\overline{\Omega}_{\mathrm{Cb}} - \Omega_{\mathrm{Cb}}(x)\right)^2} \tag{2}$$

$$M_{\mathrm{G}}(x) = \frac{1}{1 - \mathrm{e}^{-C(x)}} M_{\mathrm{CrCb}}(x) + \left(1 - \frac{1}{1 - \mathrm{e}^{-C(x)}}\right) M_{\mathrm{Y}}(x) \tag{3}$$

In (3) $C(x)$ is a coefficient, calculated in (5) depending on saturation of each pixel (note the usage of RGB space):

$$C_{\mathrm{c}}(x) = \max\left(\Omega_{\mathrm{R}}(x), \Omega_{\mathrm{G}}(x), \Omega_{\mathrm{B}}(x)\right) - \min\left(\Omega_{\mathrm{R}}(x), \Omega_{\mathrm{G}}(x), \Omega_{\mathrm{B}}(x)\right) \tag{4}$$

$$C(x) = \left(10 - 0.5 C_{\mathrm{c}}(x)\right). \tag{5}$$

Local Saliency Map (LSM). LSM is a map, calculated on the basis of idea of center-surround histograms (initially proposed in [15]), where a sliding window $P(x)$ of size p, centred over a pixel x, is compared to a surrounding area $Q(x)$, so that $(Q(x) - P(x)) = p^2$. Let a center histogram $H_{\mathrm{C}}(x)$ be a histogram of pixel intensities in window $P(x)$ (with $H_{\mathrm{C}}(x, i)$ being a value of i-th bin of this histogram, respectively), and a surrounding histogram H_{S} — a histogram of pixel intensities in area $Q(x)$. Then the center-surround feature $d_{ch}(x)$ can be calculated for each channel $ch \in \{\mathrm{Y}, \mathrm{Cr}, \mathrm{Cb}\}$ as in (6) — a normalized sum of difference over all 256 histogram bins. For pixels near the edge of image the areas $P(x)$ and $Q(x)$ are only over the image itself.

$$d_{ch}(x) = \sum_{i=1}^{255} \left(\frac{H_{\mathrm{C}}(x,i)}{|H_{\mathrm{C}}(x)|} - \frac{H_{\mathrm{S}}(x,i)}{|H_{\mathrm{S}}(x)|} \right) \tag{6}$$

$$D(x) = \frac{1}{1 - \mathrm{e}^{-C_{\mu}(x)}} d_{\mathrm{Y}}(x) + \left(1 - \frac{1}{1 - \mathrm{e}^{-C_{\mu}(x)}} \right) \max\left(d_{\mathrm{Cb}}(x), d_{\mathrm{Cr}}(x) \right) \tag{7}$$

In (6) $|H_{\mathrm{C}}|$ and $|H_{\mathrm{S}}|$ represent sum over all histogram bins. $D(x)$, an LSM, is calculated in (7) as a non-linear fusion of center-surround features $d_{ch}(x)$, with coefficient $C_{\mu}(x)$ being an average color saturation over window $P(x)$. It is calculated in (8) as average of saturation $C(x)$ on area $P(x)$ from (5).

$$C_{\mu}(x) = \frac{\sum_{k}^{P(x)} C(k)}{p} \tag{8}$$

In LSM calculation parameter p is a variable, which can be defined as $p = (WSC * w)(WSC * h)$, where w and h represent width and height of image, respectively, and WSC is window size coefficient, which can be tuned (in [18] its best value is stated as 0.4).

Final Saliency Map. Final saliency map, or just Saliency Map (SM), denoted as $M_{\mathrm{final}}(x)$, is a map resulting from a non-linear fusion of GSM and LSM as shown in (9).

$$M_{\mathrm{final}} = \begin{cases} D(x) & \text{if } D(x) > M(x) \\ \sqrt{D(x)M(x)} & \text{otherwise} \end{cases} \tag{9}$$

2.2 Analogy between "Salient Object Detection" and "Eye Fixation" Tasks

As it has been mentioned in introductory section, we have considered the "eye-fixation" paradigm as evaluation benchmark for experimental validation of investigated approach. The goal in the "eye-fixation" paradigm is modelling the human eye-fixation (supposed linking human visual attention) when watching a scene (namely an image). The main applicative goal is the prediction of humans' center of attention versus the presented image. Most applications deal with either the accurate design of web pages' visual content or relate the design of visual content for advertisement issues (commercial applications). Although focusing distinct purpose and different applications in this paper, the considered paradigm provides appealing experimental credentials linking "human-like" visual skills. In fact, stating that the human eye-fixation mechanism tends following what may be considered by the human as being relevant in visually perceived information, one may establish (or state on) some similarity between the robot's artificial vision and human's natural vision.

However, in order to establish the comparison criteria (indicators and metrics) within the considered experimental frame, we need to make a two-way

appraisal: first treating the eye-fixation paradigm as an object detection problem, and then treating the object detection algorithm as eye-fixation prediction predictor.

Eye Fixation Problem as Quasi-object Detection Problem. For such comparison we use the MIT1003 database [9], holding natural outdoor and indoor pictures as stimuli and eye fixation data encoded into grayscale fixation maps. Intensity of each pixel represents a relative eye fixation probability.

Let's define a "salient region", — a quasi-object, represented in picture with pixels, each of which holds a relative probability higher, than a threshold, to be fixated on by a human. A map of these quasi-objects can be obtained via applying an arbitrary threshold to a fixation map. If we presume, that an object from the pure object detection problem can also be defined as a salient region, we can use algorithm from [18] to detect these quasi-objects on MIT1003 images (Fig. 1 shows different salient region maps for MIT1003 images).

In this case we can compare the detected areas to salient regions, obtained from MIT1003 fixation maps. For regions detection we apply algorithm from [18] with default parameters $\{IRC = 0.2; WSC = 0.4\}$. As an evaluation we can apply usual Precision/Recall technique, as in pure object detection problems, due to similarity with classification problem [20]. Figure 2 shows Precision/Recall values for different thresholds.

The Recall value in this case can be interpreted as "how much salient regions are detectable by this algorithm" - at least 68% are detectable, and at least 20% salient regions do not represent eye fixations over detectable real salient objects. Poor Precision values represent that salient regions are less in area, than real detectable objects.

Object Detection Algorithm as Quasi-eye Fixation Algorithm. According to [13], a center-surround feature based algorithm can be used in eye fixation tasks. Let us define "eye fixation saliency map" (EFSM) as a map, which shows a distribution of possibility of each pixel to represent a part of object which draws an attention, we can say, that saliency maps, obtained via object detection algorithm, can be interpreted as EFSM, and can be compared to eye fixation maps from MIT1003 database. This work mode doesn't need steps 3–4 (segmentation map and object detection).

For evaluation of algorithm we need at least 3 metrics; as shown in [20], a pretty fair evaluation can be achieved with Area Under Curve (AUC) in Judd interpretation (AUC Judd) [10], AUC in Borji interpretation (AUC Borji) [2], and Küllback-Leibler divergence (KL-DiV) [4].

AUC Judd and AUC Borji treat the eye fixation task as a pure classification task — each pixel is classified as "salient" or "background" via application of threshold over the saliency map intensity value. Each pixel which is classified as "salient", if a corresponding pixel of ground truth fixation map is also "salient", is considered as "true positive". Correspondence "salient" against "background"

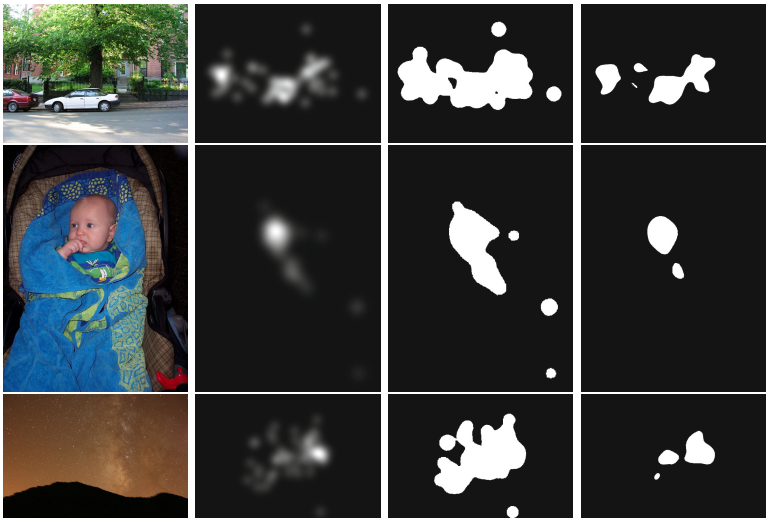


Fig. 1. Some examples of stimuli and fixation maps from MIT1003 database, and salient region maps with thresholds $T = 10$ and $T = 50$, respectively

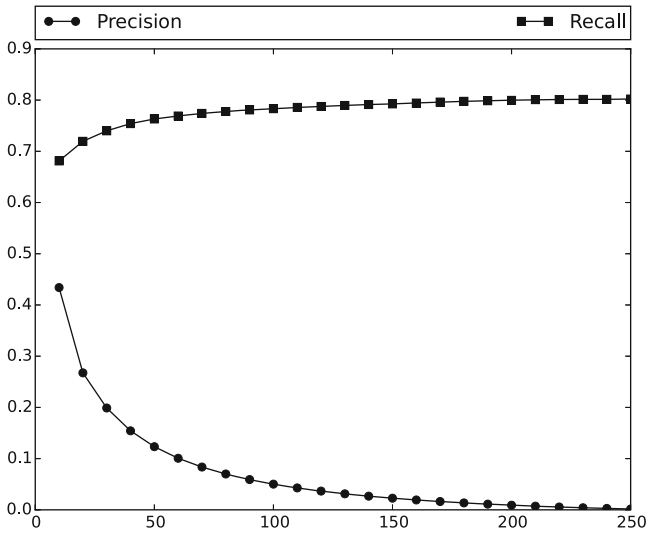


Fig. 2. Precision and Recall average values over all 1003 images for different salient region threshold values $T \in [10, 250]$.

is considered "false positive". In further text, "true positive" and "false positive" should be interpreted as stated previously.

With counting "true positive" and "false positive" pixels for each threshold applied, a receiver operating characteristic (ROC) curve [5], consisting of points in $\{TP; FP\}$ space (every point represents different threshold values), can be drawn. TP here represents a true positive rate, FP — false positive rate. In Judd interpretation [10], TP is quantity of pixels in fixation positions, considered "salient", relative to all "salient" in true fixation map pixels; FP is relative quantity of pixels in all other positions, considered "salient"; In Borji interpretation [2], FP is counted not over all, but only over the same amount of pixels, as for TP, uniformly distributed over all picture. And in both interpretation final evaluation score, AUC, is calculated as area under ROC-curve. Ideal score is 1; value 0.5 is achieved with a uniformly random saliency map, therefore, scores under 0.5 are considered as "worse, than random".

On the other hand, eye fixation task can be treated as task of distribution estimation. In this case a KL-DiV metric can be used to show dissimilarity between original and estimated distributions [4,20]. $KL_{\text{div}}(FM, SM)$ is defined over FM - original fixation map and SM - estimated saliency map, as shown in (10):

$$KL_{\text{div}}(FM, SM) = \sum_{x=1}^{X} FM(x) \log\left(\frac{FM(x)}{SM(x) + \epsilon} + \epsilon\right), \tag{10}$$

where X represents the set of all pixels, $FM(x)$ and $SM(x)$ — normalized probability distributions ($\sum_{x=1}^{X} FM(x) = 1$ and $\sum_{x=1}^{X} SM(x) = 1$), and ϵ is a small constant, used to avoid log and division by zero. Ideal score value is zero; the more is score, the bigger is dissimilarity. Table 1 shows some evaluation scores of raw saliency algorithm. We have run it with different values of $\{IRC; WSC\}$ parameters, and the best scores were achieved with different set of parameters (for example, $\{IRC = 0.25; WSC = 0.15\}$ or $\{IRC = 0.2; WSC = 0.14\}$). Yet, the default set of values $\{IRC = 0.2; WSC = 0.4\}$, as given in [18], yields scores poorer, than some other sets.

Table 1. Evaluation scores for some sets of parameters of raw saliency algorithm

IRC; WSC	AUC Judd	AUC Borji	KL-DiV
0.15; 0.1	0.6902	0.6506	0.7537
0.2; 0.14	0.7286	0.6733	0.7374
0.2; 0.4	0.7246	0.6524	0.7603
0.25; 0.15	0.7286	0.6749	0.7379

2.3 Evolutionary Optimization Based Proposed Approach

Taking into consideration the previous sections, we propose an Evolutionary Optimization-based Approach (EOA), partly based on the raw algorithm from [18]. Its steps can be shortly indexed as:

1. GSM and LSM calculation:
   (a) Calculating N LSMs with different WSC values;
   (b) Calculating GSM;
2. Fusing obtained LSMs and GSM into an "ensemble saliency map" (ESM);
3. Applying a center bias to ESM;
4. Using a threshold over biased ESM to obtain a final map;
5. Evaluating the final map through fitness function and finding the most plausible set of parameters via evolutionary process (as seen in Sect. 2.4).

Experiments with different WSC values have shown us that different sliding windows give very non-similar results for some images — while small windows give good eye fixations on small areas, bigger windows give good eye fixation prediction on either big or complex objects (examples as seen in Fig. 3). A fusion of different LSMs gives better overall results, even if they are fused in a simple linear matter, as shown in (11):

$$E(x) = w_0 * M_{\mathrm{G}}(x) + \sum_{i=1}^{N} w_i * D_i(x), \qquad \sum_{i=0}^{N} w_i = 1. \tag{11}$$

Values of weights w_i, same as quantity of LSMs (N value), are discussed in Sect. 2.4.

Next step is addition of 2D Gaussian center blob, so much discussed in eye fixation problem field (more information can be found in [1,9,10,20,24]), as shown in (12)-(15):

$$E'(x,y) = w_{\mathrm{gauss}} * G(x,y) + w_{\mathrm{nongauss}} * E(x,y) \tag{12}$$

$$G(x,y) = A * \exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2}\right) \tag{13}$$

$$\sigma_x = \frac{\mathrm{w}}{2 * k} \tag{14}$$

$$\sigma_y = \frac{\mathrm{h}}{2 * k} \tag{15}$$

Here (x, y) represent dimensional positions of each pixel in usual Cartesian 2D space, with (x_0, y_0) — geometric center of picture. Values of weights $w_{\mathrm{gauss}}, w_{\mathrm{nongauss}}$, Gaussian amplitude intensity A, and blob-size coefficient k are discussed in Sect. 2.4.

Final step is application of final threshold FT on biased ESM to reduce the blur of multi-LSM fusion and reduce the informational dissimilarity between estimated and original maps, as shown in (16):

$$SM = \begin{cases} E'(x) & \text{if } E'(x) > FT \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

2.4 Evolutionary Optimization Process

Sect. 2.3 introduces many new parameters, which have no previously found default values.

Through experiments, we found some plausible sets of values. E.g., set $IRC = 0.2$, $N = 3$, $WSC_{\text{set}} = (0.1, 0.14, 0.2)$, $W = (0.1, 0.3, 0.3, 0.3)$, $A = 128$, $w_{\text{gauss}} = 1$, $w_{\text{nongauss}} = 0.5$, $FT = 30$ gives evaluation scores: $AUC_{\text{Judd}} = 0.849$, $AUC_{\text{Borji}} = 0.731$, $KL_{\text{div}} = 0.664$.

To reach better evaluation results we used an evolutionary programming technique, where each individual represents a set of aforementioned parameters. Given a fitness function:

$$Fit = \sum_{i=1}^{N} (AJ(i) + AB(i) + KL(i)) \tag{17}$$

$$AJ(i) = \begin{cases} AUC_{\text{Judd}}(i) & \text{if } AUC_{\text{Judd}}(i) > 0.5 \\ AUC_{\text{Judd}}(i) - 1 & \text{otherwise} \end{cases} \tag{18}$$

$$AB(i) = AUC_{\text{Borji}}(i) \tag{19}$$

$$KL(i) = 1 - KL_{\text{div}}(i), \tag{20}$$

i represents one image from database ($N = 1003$ for MIT1003 and $N = 120$ for Toronto database), and $AUC_{\text{Judd}}(i)$, $AUC_{\text{Borji}}(i)$, $KL_{\text{div}}(i)$ — evaluation scores of algorithm with given parameters over i-th image. $AJ(i)$ value depends on $AUC_{\text{Judd}}(i)$ in a non-linear way due to the nature of this metric — if its value is lower than 0.5 (worse than randomly distributed saliency map), then such individual requires a fitness penalty.

3 Experimental Results

Results, shown previously, can be interpreted as moderate justification for usage of object-detection-inspired algorithm in eye fixation problem — due to difference between these tasks, at least 20% of eye fixations are undetectable by raw algorithm and average evaluation scores of it should be a priori poorer than state-of-art scores (e.g., best benchmark AUC_{Judd} values are 0.87 [9]). Yet the real results of EOA algorithm can be optimized.

With fitness function as defined in (17)–(20), an experiment over at least 40 generations of at least 50 individuals have been held with a stop condition of not improving the best fitness in 5 generations in a row. The fittest individual was used to make a comparison with state-of-art algorithms.

The best individual have been tested on MIT1003 and Toronto [3] datasets. Such choice of datasets can be explained by the nature of their images — they are general, showing different types of situations and objects, with different image quality, indoors and outdoors, etc. — which means that they are non-specialized and are aimed for general evaluation of eye-fixation algorithms. Many more

specialized datasets also exist, but they are designed for evaluation of more precisely aimed algorithms (e.g., FiWI [22] dataset for webpage saliency, EyeCrowd [8] dataset for face recognition), and should not be considered here.

For comparison we chose three leading and up-running in MIT300 benchmark algorithms, which are considered best or almost best (as described in [9,20]) via some sets of evalution metrics: eDN [26], BMS [28] and RARE2012 [21]. As a comparison metric, we use previously mentioned AUC_{Judd}, AUC_{Borji}, KL_{div} and — as their fusion — fitness function Fit, as defined in (17)–(20). Also it should be noted, that magnitude of Fit value depends on the number of images processed — i.e., for MIT1003 dataset (1003 images) values are between 1700–2000, and 200–250 for Toronto dataset (120 images).

Table 2. Evaluation scores for different state-of-art algorithms in comparison to EOA on MIT1003 and Toronto datasets.

Algorithm	Dataset	AUC_{Judd}	AUC_{Borji}	KL_{DiV}	Fitness
EOA	MIT1003	0.8422	0.7445	0.6389	1951.179
eDN	MIT1003	0.8651	0.7685	0.6681	1971.43
BMS	MIT1003	0.7652	0.6103	0.6261	1736.616
RARE2012	MIT1003	0.7706	0.6171	0.6052	1772.817
EOA	Toronto	0.8372	0.6375	0.4504	242.91
eDN	Toronto	0.8541	0.6291	0.4818	240.162
BMS	Toronto	0.7461	0.5368	0.4201	222.539
RARE2012	Toronto	0.7688	0.5381	0.391	228.91

As Table 2 shows, EOA scores are comparable with the "leading" eDN algorithm. Main difference between algorithms' outputs, which yields such results, is based on eDN maps showing higher expectation of fixation points in average (high absolute values of probability in every point, represented by higher intensity in fixation maps, as can be seen in Fig. 3), which leads to both better relative "true positive" against "false positive" rate but also higher absolute "false positive" rate. EOA maps, on the contrary, represent lower expectation level in average, which leads to lower absolute values of both "true positive" and "false positive" rates.

Therefore, while AUC_{Judd} score is better for eDN, KL_{div} score is better for EOA. AUC_{Borji} score shows a "tie" situation between both algorithms. It is also pertinent to emphasize that although eDN is slightly finer regarding the processing of the MIT dataset (reaching a fitness of 1971.43 versus 1951.18 for EOA – i.e. 1% relative ascendancy), EOA remains superior regarding the processing of the Toronto dataset (carrying out a fitness of 242.91 versus 240.16 for eDN – i.e. 1% relative supremacy).

Another key point of difference between eDN and EOA is in implementations — statistical non-learning versus deep networks learning — which gives 10 times faster performance in the same stack of technologies (software:

Python-implemented single threaded process, hardware: Intel Core i7 3,5 GHz, 1 sec. versus 10 sec. for 1024x768 image).

If comparing EOA to BMS and RARE2012, we need to take into account the difference in nature of algorithms. EOA takes its base on the object detection algorithm, and BMS and RARE2012 try to predict just the exact points of eye fixations. This leads to output difference: EOA (same as eDN) highlights, in average, more points, than BMS or RARE2012. Such behaviour leads to higher Küllback-Leibler divergence, showing higher dissimilarity between two saliency distributions (predicted by algorithm and provided by datasets). But the other two scores, AUC_{Judd} and AUC_{Borji}, show that BMS and RARE2012 results are worse than EOA and eDN.

These distinctive results between EOA and other state-of-art algorithms give us a permission to say that our algorithm can compete in the field of eye-fixation problem.

4 Conclusions and Further Work

On the one hand, statistical nature of the proposed model relating image statistical properties through the center-surround saliency, and on the other hand its evolutionary-process based parameters tuning affords its pretty general competency for detecting visual saliency in images. The aforementioned basements proffer also a comprehensive and solid theoretical background to the proposed approach. Beside these appealing points, the dual analysis of analogy between "salient object detection" and "eye-fixation" tasks (presented in Sect. 2.2) shows the fitness of the proposed approach with human's eye-fixation behaviour emerging some kind of human-like visual skill of the proposed method. This sets up a suitable skill for human-like artificial vision, opening the lane toward human-like robots? vision with proffering them kind of artificial visual attention.

Although presenting appealing skills, a number of capabilities of the presented approach remain limited and could be enhanced. One of those deficiencies in the current implementation concerns the resulted extent of the salient zone (e.g. the blur areas on salient map) which may overlap several salient areas making them considered as a unique object or may cover a number of inconsequential (non-salient) zones stirring up factual objects. Thus, the reduction of such blur zones (and therefore, "false positive" rate) will enhance the visual attention focus. An adaptive final saliency threshold (versus the statistical features of image) could overcome this inauspicious shortage. The design of such adaptive threshold constitutes short-term perspective of the present work. The second and at the same time the middle-term perspective is to reposition the "eye-fixation" (or "fixation" focus) in term of the "object-fixation" paradigm. This could be reached by adding machine-learning skills to the current system, achieving the above-mentioned human-like robots? vision along with visual attention ability.

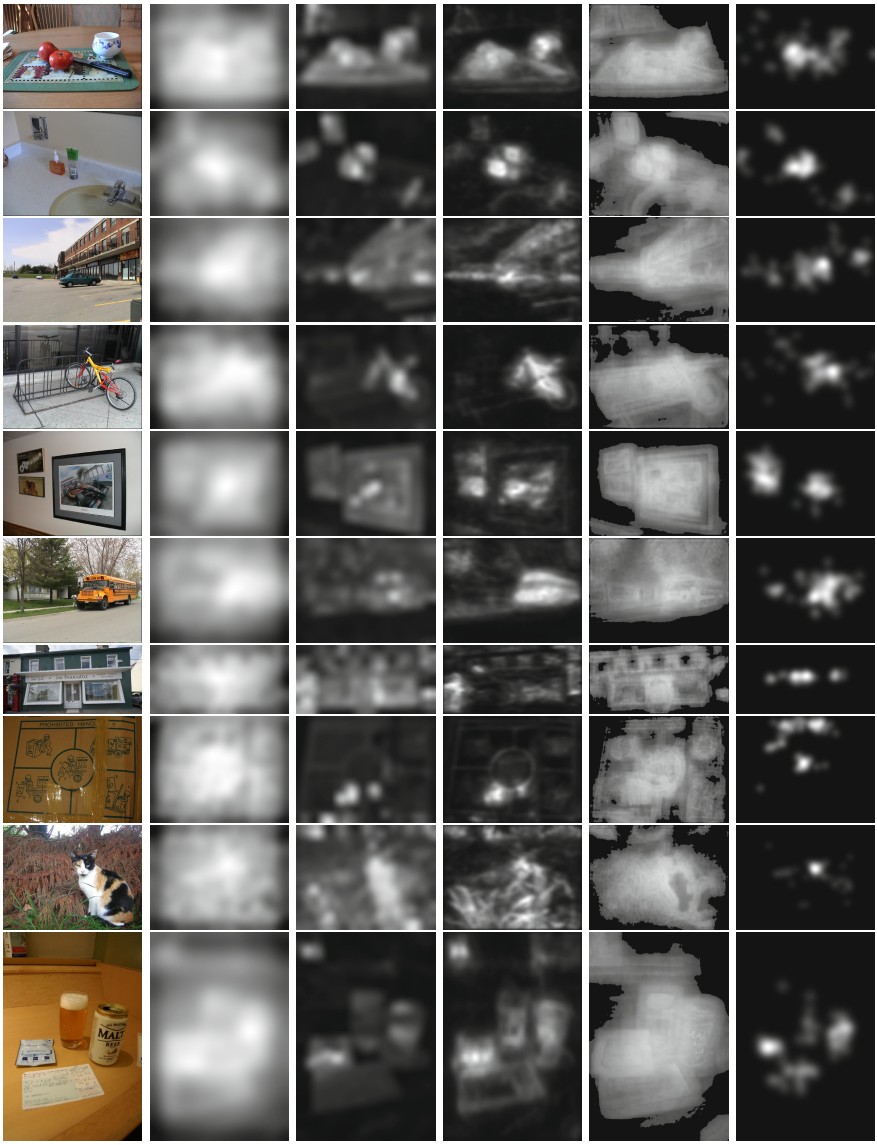


Fig. 3. Some examples of stimuli from Toronto and MIT1003 databases, estimated fixation maps as processed by eDN, BMS, RARE2012, EOA algorithms, and ground truth fixation maps.

References

1. Borji, A., Itti, L.: State-of-the-Art in Visual Attention Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(1), 185–207 (2013)
2. Borji, A., Tavakoli, H.R., Sihite D.N., Itti, L.: Analysis of Scores, Datasets, and Models in Visual Saliency Prediction. In: Proc. IEEE ICCV, pp. 921–928 (December 2013)
3. Bruce, N., Tsotsos, J.: Attention based on information maximization. J. Vision **7**(9), 950–950 (2007)
4. Contreras-Reyes, J.E., Arellano-Valle, R.B.: Küllback-Leibler divergence measure for multivariate skew-normal distributions. Entropy **14**(9), 1606–1626 (2012)
5. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters **27**, 861–874 (2006)
6. Hayhoe, M., Ballard, D.: Eye Movements in Natural Behavior. Trends in Cognitive Sciences **9**, 188–194 (2005)
7. Holzbach, A., Cheng, G.: A scalable and efficient method for salient region detection using sampled template collation. In: Proc. IEEE ICIP (2014)
8. Jiang, M., Xu, J., Zhao, Q.: Saliency in crowd. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 17–32. Springer, Heidelberg (2014)
9. Judd, T., Durand, F., Torralba, A.: A Benchmark of Computational Models of Saliency to Predict Human Fixations. MIT Technical Report (2012). http://saliency.mit.edu/
10. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to Predict Where Humans Look. In: Proc. IEEE ICCV, pp. 2106–2113 (2009)
11. Kadir, T., Brady, M.: Saliency, Scale and Image Description. J. Vision **45**(2), 83–105 (2001)
12. Kachurka, V., Madani, K., Sabourin, C., Golovko, V.: A statistical approach to human-like visual attention and saliency detection for robot vision: application to wildland fires' detection. In: Golovko, V., Imada, A. (eds.) ICNNAI 2014. CCIS, vol. 440, pp. 124–135. Springer, Heidelberg (2014)
13. Kienzle, W., Franz, M.O., Schölkopf, B., Wichmann, F.A.: Center-Surround Patterns Emerge as Optimal Predictors for Human Saccade Targets. J. Vision **9**, 1–15 (2009)
14. Koehler, K., Guo, F., Zhang, S., Eckstein, M.P.: What do saliency models predict? J. Vision **14**(3), 1–27 (2014)
15. Liu, T., Sun, J., Zheng, N.N., Shum, H.Y.: Learning to Detect a Salient Object. In: Proc. IEEE ICCV, pp. 1–8 (2007)
16. Navalpakkam, V., Itti, L.: An integrated model of top-down and bottom-up attention for optimizing detection speed. In: Proc. IEEE CVPR, pp. 2049–2056 (2006)
17. Rajashekar, U., van der Linde, I., Bovik, A.C., Cormack, L.K.: GAFFE: A Gaze-Attentive Fixation Finding Engine. IEEE Trans. Image Processing **17**(4), 564–573 (2008)
18. Ramík, D.M.: Contribution to complex visual information processing and autonomous knowledge extraction: application to autonomous robotics. Ph.D. dissertation, Université Paris-Est, Pub. No. 2012PEST1100 (2012)
19. Ramík, D.M., Sabourin, C., Madani, K.: Hybrid salient object extraction approach with automatic estimation of visual attention scale. In: Proc. IEEE SITIS (2011)
20. Riche, N., Duvinage, M., Mancas, M., Gosselin, B., Dutoit, T.: Saliency and human fixations: state-of-the-art and study of comparison metrics. In: Proc. IEEE ICCV, pp. 1153–1160, December 2013

21. Riche, N., Mancas, M., Duvinage, M., Mibulumukini, M., Gosselin B., Dutoit, T.: RARE2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. Signal Processing: Image Communication (2013), doi:10.1016/j.image.2013.03.009, issn:0923–5965
22. Shen, C., Zhao, Q.: Webpage saliency. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 33–46. Springer, Heidelberg (2014)
23. Ramanathan, S., Katti, H., Sebe, N., Kankanhalli, M., Chua, T.-S.: An eye fixation database for saliency detection in images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 30–43. Springer, Heidelberg (2010)
24. Tatler, B.W.: The Central Fixation Bias in Scene Viewing: Selecting an Optimal Viewing Position Independently of Motor Bases and Image Feature Distributions. J. Vision **14**, 1–17 (2007)
25. Triesch, J., Ballard, D.H., Hayhoe, M.M., Sullivan, B.T.: What You See Is What You Need. J. Vision **3**, 86–94 (2003)
26. Vig, E., Dorr, M., Cox. D.: Large-Scale optimization of hierarchical features for saliency prediction in natural images computer vision and pattern recognition. In: IEEE Computer Vision and Pattern Recognition (CVPR), pp. 2798–2805 (2014)
27. Võ, M.L.-H., Smith, T.J., Mital, P.K., Henderson, J.M.: Do the eyes really have it? Dynamic allocation of attention when viewing moving faces? J. Vision **12**(13), 3 (2012)
28. Zhang, J., Sclaroff, S.: Saliency detection: a boolean map approach. In: Proc. IEEE ICCV, pp. 153–160 (2013)

Towards a Shared Control Navigation Function: Efficiency Based Command Modulation

Manuel Fernández-Carmona(✉), José Manuel Peula Palacios, Cristina Urdiales García, and Francisco Sandoval

Ingeniería de Sistemas IntegradoS group, ETSI Telecommunication, University of Málaga, 29071 Málaga, Spain
{mfcarmona,jmpeula,acurdiales,fsandoval}@uma.es
http://www.grupoisis.uma.es

Abstract. This paper presents a novel shared control algorithm for robotized wheelchairs. The proposed algorithm is a new method to extend autonomous navigation techniques into the shared control domain. It reactively combines user's and robot's commands into a continuous function that approximates a classic Navigation Function (NF) by weighting input commands with NF constraints. Our approach overcomes the main drawbacks of NFs -calculus complexity and limitations on environment modeling- so it can be used in dynamic unstructured environments. It also benefits from NF properties: convergence to destination, smooth paths and safe navigation. Due to the user's contribution to control, our function is not strictly a NF, so we call it a pseudo-navigation function (PNF) instead.

Keywords: Shared control · Navigation function · Potential fields · Mixed initiative control · Power wheelchairs · Assistive robotics

1 Introduction

Europe is aging fast, nearly 20 percent of its population will be above 60 years old in 2050. [1]. The loss of functional abilities during senior age increases dependency. More specifically, loss of mobility is a functional loss with severe impact on user capabilities. Mobility plays a central role in an independent lifestyle, as enabler of many other activities. Since human and economical resources to assist persons with dependencies are not enough, enabling technologies have become very important lately.

In some cases, a robotic wheelchair is needed to increase the user's independence. Most these wheelchairs follow the shared control paradigm. Shared control addresses human-robot interaction in navigation. There are two basic criteria to model robot and user interaction in shared control: i) safeguarded operation (the user is in charge unless a potentially dangerous situation is detected [2]); and ii) shared control (the robot performs specific maneuvers on user's demand or by automatic triggering). Shared control maneuvers may range from specific tasks

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 185–198, 2015.
DOI: 10.1007/978-3-319-19258-1_16

such as "cross door" or "follow corridor" [3,4] to fully autonomous navigation to designed targets [3,5].

Shared control navigation traditionally results into more or less abrupt target and trajectory changes. Users also disfavor these discontinuities. Medical doctors don't support these approaches either because robots always perform all challenging tasks alone and hence, residual user abilities decay on time [6]. Our shared control algorithm -collaborative control- addresses these discontinuity problems.

We propose a new shared control algorithm -collaborative control- based on Navigation Functions (NF). NF is a cost function for the optimization paradigm, first described in [7]. The NF's properties guarantee smooth convergence to an unique minimum in the space where they are defined. This paper describes how Collaborative Control approaches human-robot collaboration to a NF based autonomous control as much as possible. Human intervention introduces unpredictable variations, making impossible to obtain a NF from them in all situations.

The paper is structured as follows. First, we present the state of the art in shared control. Then we describe the theoretical model of collaborative control and its benefits. The next section describes mathematical model linking NF and collaborative control.

An improved version of collaborative control, is depicted in Section 5. Then, we use a set of experiments to test validity of these improvements. Finally, we present the conclusions of this work.

2 Related Work

The main task of this work is to help wheelchair users to navigate in a safe and comfortable way. Autonomous robots typically solved navigation through two different approaches: i) deliberative algorithms (a high level model uses sensory information to identify current situation and/or generate a response from a discrete set [8,9]), ii) reactive algorithms (an equation links sensory information with motor responses, using physic models or insect-alike behaviors [10,11]).

Optimization techniques belong to this latter approach: we minimize an scalar cost function using negative descend gradient. Generated vector field describes an optimal path to a destination. The scalar function can be obtained using optimal methods such as wavefront propagation algorithms [12] and shortest path algorithms [13] or heuristic methods like maximum clearance algorithms [14]. Potential Fields Approach (PFA) is also a family of feedback motion planning algorithms inspired in obstacle avoidance methods. PFA overlaps several artificial potential fields: an attractor potential field to target and repulsor potential fields to avoid obstacles. They are a simple version of cost functions, suitable for fast path recalculation but not very accurate or stable [15]. Local minima is another PFA mayor limitation. We originally proposed a continuous and implicit shared control algorithm -collaborative control- based on obstacle avoidance methods [16]. Collaborative control extends autonomous navigation algorithms into the shared control domain.

Collaborative control combines user and robot commands into an emergent command in a reactive way. More specifically robot commands are generated with a modified version of PFA [17]. This technique can combine several goals and constrains in a continuous way, e.g., user intentions could be modeled as an additional target into the potential field.

3 Collaborative Control in Navigation Functions

Collaborative control tries to extend NFs into the shared control domain. Thus, users benefit from a smooth navigation, headed for designated target. This algorithm combines user and robot commands into a function with the same properties as the NF gradient.

In our approach, users receive assistance on a need basis: the worse they perform, the more help they get. This improves user's acceptance and smooth control discontinuities, that affect both user and robot.

An NF φ is a $\mathbb{C}^2$ (at least differentiability class 2) artificial potential function (PF) on a compact manifold M such that $\varphi : M \longrightarrow [0, 1]$. Unlike other PF, it encodes a goal $\bar{q}_g$ as the unique global minimum, $\varphi(\bar{q}_g) = 0$, and achieves a maximum of 1 on the entire boundary of M , i.e. $\varphi(\partial M) = 1$ without any divergence. Once constructed, one can obtain a second order autonomous controller which is guaranteed to converge to $\bar{q}_g$ for all points in M, by simply following the negative gradient of the NF [18].

These functions are hard to implement for dynamic environments. This is even more challenging in a real time system. Hence, they have serious limitations for unstructured environments [19]. Furthermore, NFs have severe limitations within the shared control paradigm, because commands provided by the users won't typically fit the NF properties.

In our case, any collaborative motion vector described by $\boldsymbol{V}_C$ includes a component provided by human $\boldsymbol{V}_H$ through any input device (typically a joystick) and a PFA command $\boldsymbol{V}_R$ as robotic counterpart. This approach was originally proposed in [20].

This solution, though, splits control equally between user and machine. This is not always wise: some users could perform dangerous maneuvers or generate erratic commands. Others might be completely unable to produce an adequate command to solve some specific situation. Indeed, this solution is discontinuous. Convergence issues derive from using PFA: convergence is compromised, as we have included a non analytic component ($\boldsymbol{V}_H$).

To cope with this issue we evaluated the local efficiency of commands $\boldsymbol{V}_R$ and $\boldsymbol{V}_H$. Both vectors were weighted so that control transitions are smoother. This overcomes the problem of convergence and most PFA limitations. It also improves continuity in the emergent command. Now, commands are weighted using an efficiency term (η). η is a local, memoryless, reactive metric without any temporal information like curvature or trajectory length. PFA commands only depend on current state $\bar{q}_i$, i.e. they are purely reactive. Consequently, η must also be obtained at reactive level, and based uniquely on local performance.

$$\boldsymbol{V}_C = \ \eta(\bar{q}_i, \boldsymbol{V}_R)\ \boldsymbol{V}_R + \ \eta(\bar{q}_i, \boldsymbol{V}_H)\ \boldsymbol{V}_H \tag{1}$$

In our approach η weights motion commands according to their likeness to a NF. η has one component per prerequisite of a NF. Rimon and Koditschek defined in [7] the properties of a NF. It must be:

1. smooth (at least in $\mathbb{C}^2$)
2. have a single local minimum at $\bar{q}_g$
3. maximal along the boundaries of the domain Q
4. a Morse function (its critical points are non-degenerate)

The fourth prerequisite is ensured in our work by the decomposition of the space into simpler convex subspaces, as proposed in [21], so it has no impact on η. Consecuently, we define η as the average of only three factors, one per each NF property.

$$\eta(\bar{q}_i, \boldsymbol{V}) = \frac{\eta_{sm}(\bar{q}_i, \boldsymbol{V}) + \eta_{dir}(\bar{q}_i, \boldsymbol{V}) + \eta_{sf}(\bar{q}_i, \boldsymbol{V})}{3} \tag{2}$$

By including η, into (1), we reinforce the NF conditions, as will be shown in the following subsections.

3.1 Efficiency Factor: Smoothness

Smoothness η_{sm} penalizes sharp direction changes in input $\boldsymbol{V}$. This corresponds to the first NF requisite: smoothness at least in $\mathbb{C}^2$. Any value implying a discontinuity in motion due to a sharp direction change will be penalized by this factor. It is proportional to the angle between the vehicle heading in position $\bar{q}_i$ and proposed command $\boldsymbol{V}$ (α_{sm}), as described in (3). We use a positive constant C_{sm} to adjust smoothness influence on η (see Fig. 1).

$$\eta_{sm} = e^{-C_{sm} \cdot |\alpha_{sm}|} \tag{3}$$

3.2 Efficiency Factor: Directness

Directness η_{dir} is related to the orientation with respect to the target. A NF must be *polar*, i.e. it should have an unique minimum, located at $\boldsymbol{q}_G$. The gradient descent of such a function will always approach that minimum. Once again, it is impossible to guarantee that any command will follow this rule, but it is possible to reward such a behavior.

The angle formed by the proposed motion command and the target direction is an obvious local indicator. This angle (α_{dir}) should be low in a polar function: hence Eq. 4 will be maximum for angle 0°. Positive constant C_{dir} fixes the impact of η_{dir} on η.

$$\eta_{dir} = e^{-C_{dir} \cdot |\alpha_{dir}|} \tag{4}$$

3.3 Efficiency Factor: Safety

Safety η_{sf} penalizes commands pointing towards obstacles, and, in general, commands bringing the mobile closer to obstacles. A NF must be maximum and uniform in the boundaries of space Q, conformed by the environment obstacles. We should have maximum potential on those boundaries, and its gradient should be normal to it. The gradient of the repulsory field defined by a PFA generates a vector field pointing opposite to nearest obstacle. The angle between the opposite to this vector field and the proposed command (α_{sf}) measures our deviation from a NF. The lower α_{sf} is, the safer and the more similar to a NF the command is. The absolute distance to nearest obstacle d_{obst} is also taken into account, divided by the maximum measurable distance d_{max}. This correction was introduced to reduce the influence of this factor with the distance. We also use a positive constant C_{sf} to avoid dominance of η_{sf} over other factors.

$$\eta_{sf} = 1 - e^{-C_{sf} \cdot |\alpha_{sf} + \frac{d_{obst}}{d_{max}}|} \tag{5}$$

All necessary elements for η calculation are summarized in Fig. 1.

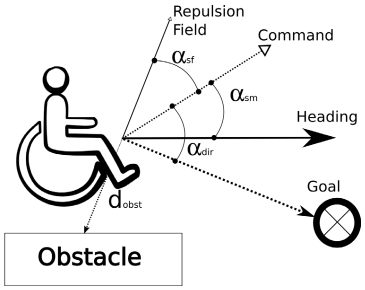


Fig. 1. η calculation parameters

C_i constants are used to take into account very specific tests environments or wheelchair structures, though in general they can be set to 1. For example, a wheelchair with a high center of mass should have a high C_{sm} to prevent instability. A crowded environment requires high C_{sf} but lower d_{dmax} to properly shape η_{sf}. Their initial values must change to fit the scenario. However, these constants can be easily chosen for a standard power wheelchair operating in indoor environments.

4 Collaborative Control as a Navigation Function

In this section we will analyze the link between our collaborative control and the properties of a generic navigation function $f(\bar{x})$ as defined in [7]. Our collaborative command will be compared with the divergence of a navigation function,

i.e. $\boldsymbol{V}_C = \boldsymbol{\nabla} f(\bar{x})$. Divergence operator is a linear and differential operator. We will use the cartesian two dimensional divergence operator: $\boldsymbol{\nabla} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$.

Commands, $\boldsymbol{V}(\bar{x}, M)$, depend on the current mobile position $\bar{x}$ and surrounding map M. A command efficiency, $\eta(\boldsymbol{V}(\bar{x}, M), \bar{x}, M)$, also depends on the current position and map, but we will denote it just as $\eta(\boldsymbol{V}(\bar{x}))$. Efficiency is a monotonic function at any given point $\bar{x}$, with image [0,1).

4.1 Property: Smoothness

We will apply divergence operator to the smoothness definition:

$$\lim_{\bar{x}\to\bar{C}^+} f(\bar{x}) = \lim_{\bar{x}\to\bar{C}^-} f(\bar{x}) \tag{6}$$

$$\begin{aligned}
&\nabla \lim_{\bar{x}\to C^+} f(\bar{x}) = \lim_{\bar{x}\to\bar{C}^+} \nabla f(\bar{x}) = \\
&= \lim_{\bar{x}\to\bar{C}^+} \eta(\boldsymbol{V}_H(\bar{x})) \lim_{\bar{x}\to\bar{C}^+} \boldsymbol{V}_H(\bar{x}) + \\
&+ \lim_{\bar{x}\to\bar{C}^+} \eta(\boldsymbol{V}_R(\bar{x})) \lim_{\bar{x}\to\bar{C}^+} \boldsymbol{V}_R(\bar{x})
\end{aligned}$$

Efficiency and robot commands are *continuous*. Their values converge when they approach to a point C. Thus, smoothness depends on the product of human commands and its efficiency. We cannot guarantee continuity in human commands. However, our efficiency function compensates this, decreasing their impact on the equation when discontinuities appear so that smoothness is mostly granted by the robot commands.

4.2 Property: Polar Function

$$f(\bar{x}) \geq f(\bar{q}_g) \tag{7}$$

A navigation function must have one single minimum at destination $\bar{q}_g$. Necessary minimum conditions are:

- First partial derivative is equal to zero at the critical point.
- Second partial derivatives are positive.

$$\nabla f(\bar{q}_g) = \bar{0} \tag{8}$$

$$\begin{aligned}
f_{xx}(\bar{q}_g) &\geq 0 \\
f_{yy}(\bar{q}_g) &\geq 0
\end{aligned} \tag{9}$$

We can expand Eq. 8 using collaborative control definition as:

$$\begin{aligned}
\nabla f(\bar{q}_g) = \eta(\boldsymbol{V}_H(\bar{q}_g))\boldsymbol{V}_H(\bar{q}_g) + \\
+ \eta(\boldsymbol{V}_R(\bar{q}_g))\boldsymbol{V}_R(\bar{q}_g)
\end{aligned}$$

Any command at $\bar{q}_g$ pointing out of it will have a very low efficiency. The only command in this situation with high efficiency is $\bar{0}$, thus this requisite is fulfilled in any case.

We also apply collaborative control definition to Eq. 9, resulting:

$$f_{x_i x_j}(\bar{q}_g) = \frac{\partial f(\bar{q}_g)}{\partial x_i x_j} = \frac{\partial \nabla_{x_j} f(\bar{q}_g)}{\partial x_i} =$$
$$= [\frac{\partial}{\partial x_i}\eta(\boldsymbol{V}_H(\bar{q}_g))]V_{Hx_j}(\bar{q}_g) + \eta(\boldsymbol{V}_H(\bar{q}_g))[\frac{\partial}{\partial x_i}V_{Hx_j}(\bar{q}_g)]+$$
$$[\frac{\partial}{\partial x_i}\eta(\boldsymbol{V}_R(\bar{q}_g))]V_{Rx_j}(\bar{q}_g) + \eta(\boldsymbol{V}_R(\bar{q}_g))[\frac{\partial}{\partial x_i}V_{Rx_j}(\bar{q}_g)]$$

Efficiency has a maximum at $\bar{q}_g$, so:

$$\frac{\partial}{\partial x_i}\eta(\boldsymbol{V}(\bar{q}_g)) \to 0$$
$$f_{x_i x_j}(\bar{q}_g) = \eta(\boldsymbol{V}_H(\bar{q}_g))[\frac{\partial}{\partial x_i}V_{Hx_j}(\bar{q}_g)]+$$
$$+ \eta(\boldsymbol{V}_R(\bar{q}_g))[\frac{\partial}{\partial x_i}V_{Rx_j}(\bar{q}_g)] \tag{10}$$

This factor always tend to zero. A highly variant command will have low efficiency. Highly efficient commands must have low variation and value near $\bar{0}$.

4.3 Property: Maximal at Frontiers

This condition is similar to previous one:

$$f(\bar{x}) \leq f(\bar{q}_o) \tag{11}$$

A navigation function must be maximum at borders $\bar{q}_o$. This point must be a critical point, so necessary conditions are:

- First partial derivative is equal to zero at the critical point.
- Second partial derivatives are negative.

These can be expressed as:

$$\nabla f(\bar{q}_o) = \bar{0} \tag{12}$$
$$f_{xx}(\bar{q}_o) \leq 0$$
$$f_{yy}(\bar{q}_o) \leq 0 \tag{13}$$

We can expand these equations using previous results:

$$\nabla f(\bar{q}_o) = \eta(\boldsymbol{V}_H(\bar{q}_o))\boldsymbol{V}_H(\bar{q}_o) + \eta(\boldsymbol{V}_R(\bar{q}_o))\boldsymbol{V}_R(\bar{q}_o)$$
$$f_{x_i x_j}(\bar{q}_o) = \eta(\boldsymbol{V}_H(\bar{q}_o))[\frac{\partial}{\partial x_i}V_{Hx_j}(\bar{q}_o)]+$$
$$+ \eta(\boldsymbol{V}_R\bar{q}_o)[\frac{\partial}{\partial x_i}V_{Rx_j}(\bar{q}_o)]$$

Efficiency properties guarantee these restrictions. Efficiency has a minimum at $\bar{q}_o$, which implies:

$$\eta(\boldsymbol{V}(\bar{q}_o)) \to 0$$
$$\frac{\partial}{\partial x_i}\eta(\boldsymbol{V}(\bar{q}_o)) \to 0$$

This is, all second derivatives are zero at $\bar{q}_o$.

4.4 Summary

Relationship between navigation functions and efficiency factor can be summarized with these requisites, provided by efficiency factors:

- Discontinuities are compensated by efficiency function
- Efficiency has a single minimum at $\bar{q}_o$
- Efficiency has a maximum at $\bar{q}_g$

However, this collaborative control solution still has some drawbacks: i) It is purely reactive: it won't prevent potentially dangerous situations nor recall previous hazardous situations. It just avoids them with the information at hand in a given time instant. ii) User's commands have no specific predominance over the robot's ones or viceversa: user's and robot's commands will have always a relative impact. iii) Users should override control when they are performing remarkably well, to avoid frustration and reward their effort, as commented.

It needs to be noted that human commands are typically hard to predict, specially for persons with different disabilities. Similarly, a dynamic environment is not that predictable either. However, we can predict up to some point a given user's efficiency coping with a given environment structure if we analyze how well this person has driven since the last significant change. This can be used to provide some inertia to the proposed collaborative navigation technique and, hence, overcome the commented limitations in an attempt to approximate better a NF.

5 Modulated Collaborative Control

The previous section concluded with the limitations on purely reactive user's and robot's command combination. Our present target is to make emergent commands fulfill NF properties. We introduce a new factor K in Eq. 1 to modulate user-computer control ratio. This factor works like a *wave-envelope* to provide efficiency dependent inertia, i.e. people who consistently drive well at a given environment area receive more control despite punctual errors. A similar concept was successfully introduced in [22] to include user's biometric information into collaborative control. K depends now on the average η_H in a recent time interval. This time interval starts at the last sharp discontinuity point of η_H,

to take into account that persons may drive differently at different areas of the environment depending on their specific disability. High $\bar{\eta}_H$ values will provoke high K values. Emergent commands are now generated using Eq. 14.

$$\boldsymbol{V}_C = (1 - K(\bar{\eta}_H))\ \eta_R\ \boldsymbol{V}_R + K(\bar{\eta}_H)\ \eta_H\ \boldsymbol{V}_H \tag{14}$$

K is a discrete step function that depends on $\bar{\eta}$.

The more user's performance decreases, the more assistance is provided. According to the PCM model [23], users accept this approach better because it prevents abrupt control changes. We give more control to the robot when the user clearly performs poorly and viceversa. User will be rewarded more control by sustained efficient driving. This inertia intends to skip punctual issues and focus on area trends. Users will also perceive consistent control increases and feel a higher degree of control.

Table 1. NF likehood of navigation commands

Navigation command	η difference (%) respect to a NF algorithm				Command angle average difference (^{o})	
	η_{sm}	η_{dir}	η_{sf}	η	Mean	Std
Autonomous erratic user	22.54	37.93	28.73	29.46	-10.75	113.42
Autonomous PFA	4.13	10.51	-3.30	3.78	-0.84	46.33
Collaborative control	18.64	20.30	0.12	12.92	-1.62	50.24
Modulated Collaborative control	15.49	18.13	-3.53	9.98	3.74	46.68

Our first proposal for $K(\bar{\eta}_H)$ function was a direct proportionality. However doctors recommended us to use a discrete set of K values instead because they did not feel η can be measured with precision. Hence we defined four cases: i) warning, ii) bad performance, iii) average driving, iv) good performance. Using data from previous collaborative control tests [24], we identified those cases as: i) unfinished tests, ii) low performance tests, iii) average tests and iv) healthy user tests. We took as $\bar{\eta}_{Hi}$ the average η_H value of each identified case.

Table 1 summarizes a comparison between different control modes. It covers the difference in η between the reference NF and the different solutions presented in this section. There is a large difference between the simulated erratic user and the NF. This was to be expected, since we were simulating a low performance user. PFA is very close to NF in terms of efficiency, and even better in terms of safety. This was expected: both are optimization algorithms and PFA has sharp discontinuities in obstacle proximity -note the lower directness.

We simulated navigation of an erratic user assisted by PFA using collaborative control. Collaborative control is closer to NF than the dangerous user alone in terms of η. Our simulated user, using collaborative control, has increased his performance from 4% through 28%.

More important, safety is clearly improved. Finally, modulated collaborative control makes some improvement (about 3%) over these benefits. Modulated

collaborative control is a more flexible algorithm than the previous one. The main target of modulated collaborative control is to meet as much as possible the properties of a NF, not to achieve the highest possible efficiency.

We also provide in Table 1 statistics of the command angle difference. PFA commands are very similar to NF ones. Collaborative control combines these two sources of information. This command preserves the resemblance with a NF (less than 2 degrees of difference and almost the same deviation) despite of the user intervention. Modulated collaborative control is less precise than collaborative control (see higher angle difference). However, modulated collaborative control provides commands more coherent with NF commands (see standard deviation decrement). Modulated collaborative control commands are a little bit biased, but more similar to the NF ones.

6 Experimental Results

Our modulated collaborative control system has been tested by real users in a controlled scenario in Rome, at Fondazione Santa Lucia concept House "Casa Agevole". The key idea was improving wheelchair usability within the house since we had performed there previous tests in basic collaborative mode [24].

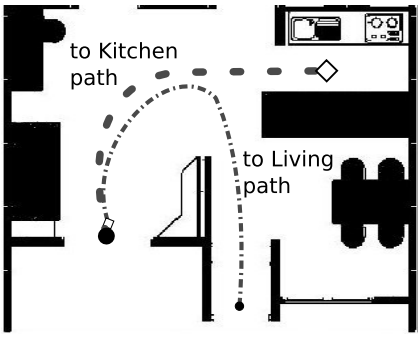


Fig. 2. Proposed path examples: to kitchen, and to living room

6.1 Experimental Set-up

Experiments consisted of accomplishing two different trajectories (Fig. 2). According to clinicians, these trajectories are frequent within the Activities of Daily Living (ADL) in a house: going from the entrance to the living room and going from the living room to the kitchen. They included narrow corridors, a great challenge both for PFA systems and novel drivers.

Experiments were performed by 10 users. Most of them had mild physical disabilities, pointed by a Barthel index over 70% and a IADL over 4 points. Cognitive scales also indicate a moderate degree of dependence.

Experiments were made using different motion planning systems in both trajectories:

– Collaborative control: This test was based on Eq. 1. It provided a reference for benchmarking as shown in Section 6.3.
– Modulated collaborative control: This experiment showed the effect of dynamically changing the amount of user control. It also showed how the variation of K is perceived by user.

Table 2. Test average results

Test type		Collaborative control		Modulated collaborative control	
Destination		Living	Kitchen	Living	Kitchen
PFA efficiency (%)	η_{sm}	53.47	70.46	50.38	60.11
	η_{dir}	67.36	74.49	65.03	75.36
	η_{sf}	94.22	97.34	91.91	97.81
	η	71.64	80.74	69.09	77.77
User efficiency (%)	η_{sm}	73.08	64.97	65.17	82.83
	η_{dir}	50.06	38.00	38.76	59.62
	η_{sf}	96.30	97.43	97.52	97.13
	η	73.04	66.78	67.08	79.73
Collaborative efficiency (%)	η_{sm}	68.41	75.09	69.35	73.97
	η_{dir}	46.26	54.13	54.33	54.29
	η_{sf}	97.82	98.29	98.00	97.71
	mean	751.87	971.70	893.00	897.48
	std	1239.97	1120.42	1806.22	1751.11
Distance to targets (mm)	mean	233.69	269.58	235.43	260.04
	std	262.31	265.81	242.14	226.73
Angle difference (°)	mean	2.38	11.98	6.48	3.72

6.2 First Experiment: Collaborative Control Navigation

Results for all experiments are summarized in Table 2. There is one column for each proposed trajectory and control approach, showing the average results. η should be a measure of how similar the navigation command is to one generated by NF. We have included another navigation performance parameter: distance to target. Real systems define not a point, but a region surrounding a target. The lower this distance is, the better the generated trajectories will be.

Emergent control commands mimic user's commands when they are efficient. Users believe they are in control most of the time. Control switches are also intuitive for users, who recover control shortly after alignement. All users achieved their final target with relative easiness.

In order to prove that the proposed profile is representative, we will compare these results with collaborative control tests. Since collaborative control is expected to adapt to each person i.e. to improve residual skills and fill in for lacking ones, resulting profiles should be closer to the benchmark profile in collaborative mode, despite the user's condition.

6.3 Second Experiment: Modulated Collaborative Control Navigation

We used the same trajectories in the previous experiments to test the new approach in Section 5. User commands will be combined again with a robotic control algorithm. Relative impact of each command is not fixed, now it is given by parameter K.

Modullated collaborative control matches average η values with K values. Thus, a user with a sustained efficiency above 50% will have a higher amount of control, a 70% using collaborative control.

On the other hand, users driving with low efficiency, e.g. crossing a narrow door, will have less control than in the previous experiment. Average K percentage value of these tests was slightly higher than fixed K: 58.87% in kitchen tests and 56.53 in living room tests.

In general, modulated control is similar to collaborative control as proposed in 6.2. Indeed, there is no major efficiency variation in average η. Collaborative control efficiency is mostly defined by user's navigation skills. Modulated collaborative control keeps the main role of the user.

However, there are very significant efficiency increases at specific locations, specifically at areas where user or robot perform particularly poorly. The main advantage of modulated collaborative control is, consequently, a better adaptation to the user, providing information about a reliable control proportion. With this approach, we obtain longer tracking periods, perceived by user as a better control. Users reported to feel a better system control with this system approach.

7 Conclusions and Future Work

In this paper, we have presented a method to extend navigation functions into human-computer shared navigation. Navigation functions offer optimal and unique solution to the autonomous navigation problem. Our shared control algorithm, collaborative control, has been defined in terms of navigation function properties. User commands are rated in collaborative control using η function.

This work has also presented an improved collaborative control algorithm. It has several benefits from its original version. First of all, it is self configurable. Now it is not necessary to fine tune the amount of user control, stated as K constant. Modulated collaborative control system changes autonomously the instant K value, according to user performance.

Finally we have presented a comparison test between original and modulated collaborative control. Modulated collaborative control preserves user predominance in navigation, adapting control in critical situations. This adaptation is not perceived by users as negative. They have, on average, the same performance.

From the user's point of view, navigation is smoother than previous version. The amount of control finally given to users reflex can be also a useful information about their state. One of the main objectives of this navigation scheme is to

make user effort at her/his best, while being intuitive. The users perceived the control adaptation, and reported to ease navigation.

Further work will be made on improving these two factors: usability and adaptability. User interfaces able to provide extra information, such as force feedback joysticks will included to reinforce feedback. Adaptability can be explored using new ways of changing K in terms of η. We have seen how η improves the convergence and stability of the resulting motion command. This allows to include different approaches within the system further than PFA.

Acknowledgments. This work has been partially supported by the Junta de Andalucia, SIAD Project No. TIC-3991 and by the Spanish Ministerio de Educacion y Ciencia (MEC), Project n. TEC2011-29106. The authors would also like to acknowledge *Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech for their support.*

References

1. Vienna Institute of Demography: The european demographic data sheet 2008 (2008)
2. Taha, T., Miro, J., Dissanayake, G.: Pomdp-based long-term user intention prediction for wheelchair navigation. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 3920–3925 (2008)
3. Parikh, S.P., Grassi Jr., V., Kumar, V., Okamoto Jr., J.: Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. IEEE Intelligent Systems **22**, 33–41 (2007)
4. Horiguchi, Y., Sawaragi, T.: Effects of probing to adapt machine autonomy in shared control systems. In: Johnson, M., Chen, P., publication co-chairs (eds.) Proc. International Conference on Systems, Man and Cybernetics, vol. 1, pp. 317–323. IEEE Service Center, Hawaii (2005)
5. Mandel, C., Huebner, K., Vierhuff, T.: Towards an autonomous wheelchair: Cognitive aspects in service robotics. In: Melhuish, C. (ed.) Proceedings of Towards Autonomous Robotic Systems (TAROS 2005), pp. 165–172. ARS Press, London (2005)
6. Kane, R.: Long-term care and a good quality of life: bringing them closer together. The Gerontologist **41**, 293–304 (2001)
7. Rimon, E.: Exact robot navigation using artificial potential functions. PhD thesis, New Haven, CT, USA, Adviser-Koditschek, Daniel E (1990)
8. Ogren, P., Leonard, N.: A convergent dynamic window approach to obstacle avoidance. IEEE Transactions on Robotics **21**, 188–195 (2005)
9. Minguez, J., Montano, L.: Nearness diagram navigation (nd): a new real time collision avoidance approach. In: Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2000, vol. 3, pp. 2094–2100 (2000)
10. Masoud, A.: Solving the narrow corridor problem in potential field-guided autonomous robots. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, pp. 2909–2914 (2005)

11. Ganapathy, V., Jie, T., Parasuraman, S.: Improved ant colony optimization for robot navigation. In: 2010 7th International Symposium on Mechatronics and its Applications (ISMA), pp. 1–6 (2010)
12. Al-Jumail, A., Leung, C.: Wavefront propagation and fuzzy based autonomous navigation (2005)
13. Dial, R.B.: Algorithm 360: shortest-path forest with topological ordering [h]. Commun. ACM **12**, 632–633 (1969)
14. Buck, S., Weber, U., Beetz, M., Schmitt, T.: Multi-robot path planning for dynamic environments: a case study. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 1245–1250 (2001)
15. Ren, J., McIsaac, K., Patel, R.: Modified newton's method applied to potential field-based navigation for mobile robots. IEEE Transactions on Robotics **22**, 384–391 (2006)
16. Urdiales, C., Poncela, A., Sanchez-Tato, I., Sandoval, F.: Efficiency based reactive shared control for collaborative human/robot navigation. In: Proc. of the IEEE Conference on Intell. Robots and Systems, IROS 2007, San Diego, USA (2007)
17. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research **5**, 90–98 (1986)
18. Cowan, N., Lopes, G., Koditschek, D.: Rigid body visual servoing using navigation functions. In: Proceedings of the 39th IEEE Conference on Decision and Control, vol. 4, pp. 3920–3926 (2000)
19. Pimenta, L., Fonseca, A., Pereira, G., Mesquita, R., Silva, E., Caminhas, W., Campos, M.: On computing complex navigation functions. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, pp. 3452–3457 (2005)
20. Poncela, A., Urdiales, C., Perez, E., Sandoval, F.: A new efficiency-weighted strategy for continuous human/robot cooperation in navigation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans **39**, 486–500 (2009)
21. Choi, W., Latombe, J.C.: A reactive architecture for planning and executing robot motions with incomplete knowledge. In: IROS 1991: Proceedings of International Conference on Intelligent Robots and Systems, pp. 24–29 (1991)
22. Urdiales, C., Fernandez-Espejo, B., Annicchiaricco, R., Sandoval, F., Caltagirone, C.: Biometrically modulated collaborative control for an assistive wheelchair. IEEE Transactions on Neural Systems and Rehabilitation Engineering **18**, 398–408 (2010)
23. Lawton, M.: Competence, environmental press, and the adaptation of older people. In: Windley, P., Byerts, T., Lawton, M.P. (eds.) Aging and the Environment: Theoretical approaches. Springer (1982)
24. Urdiales García, C., Peula Palacios, J.M., Fernández Carmona, M., Barrue, C., Pérez Rodríguez, E.J., Sánchez Tato, M.I., del Toro Lasanta, J.C., Galluppi, F., Cortes, U., Annichiaricco, R., Caltagirone, C., Sandoval Hernández, F.: A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation. Autonomous Robots (2010)

AMiRo: A Mini Robot for Scientific Applications

Thomas Schöpping(✉), Timo Korthals,
Stefan Herbrechtsmeier, and Ulrich Rückert

Cognitronics and Sensor Systems, Bielefeld University, Inspiration 1,
33619 Bielefeld, Germany
{tschoepp,tkorthals,sherbrec,rueckert}@cit-ec.uni-bielefeld.de
http://www.ks.cit-ec.uni-bielefeld.de/

Abstract. The *Autonomous Mini Robot* (AMiRo) is a modular and extensible mini robot platform, designed for scientific research and education. Its decentralized architecture enables to easily add or remove functionalities as required for any application. A well defined physical and electrical interface offers the possibility to design new modules with minimal effort. The open-source software framework for the AMiRo is already growing, since the robot is commonly used for research, education, and competitions. Several demonstrations of the system are given, which present its capabilities. Starting with a fuzzy controller for line following, these demonstrations include remote controlling as well as an implementation of an artificial neural network running on the platform.

Keywords: Research robot · Education robot · Edutainment · Minirobots · Mobile robots

1 Introduction

Small robots became a famous and affordable platform for autonomous systems in research, education, and as commercial products [10]. In contrast to large systems like the PR2 [3], which are usually equipped with high computing power and cutting edge sensors and actuators, mini robots are rather limited in their capabilities, while their functional properties barely differ. Due to their affordability, simple design, and fast setup time they are ideal for any use case.

Since the Khepera I, which was the first mini robot available for purchase [8] [5], systems have evolved and numerous platforms have been developed. On the one hand, there are very cheap robots like the Thymio II [9] that fit best to build up robot swarms. On the other hand, more powerful but also more expensive systems like the Khepera IV [1] can be used for single or multi robotics. So far there is no mini robot platform available, which combines an affordable price, powerful hardware, and free extensibility. A comparison of the mentioned platforms is given in table 1.

The goal for the *Autonomous Mini Robot* (AMiRo) was to create a system, which packs the latest off-the-shelf microelectronics and sensor technologies so it can act fully autonomous and in an energy efficient manner. It provides a

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 199–205, 2015.
DOI: 10.1007/978-3-319-19258-1_17

Fig. 1. Basic platform (left) and uncovered with extensions (right)

Table 1. Comparison of the basic properties of the Thymio II, the Khepera IV, and the AMiRo

property	Thymio II	Khepera IV	AMiRo
processing	16-bit MCU	MCU low-power SoC	3× 32-bit MCU low-power SoC* FPGA*
battery	*unspecified*	3400 mAh (5-7 hours)	2× 1950 mAh (up to 20 hours)
extensibilty	yes (restricted)	yes (proprietary)	yes (unrestricted)
target scope	home & education	research	science & education
open-source	hardware (full) software (partially)	software (partially)	software (full)

* can be added by extension modules

modular structure that can be extended as the application requires and still stays in an affordable price range. After developing a first prototype [6], the AMiRo finally became a mature device, which is already commonly used for education, scientific research, and competitions. With its modular and extensible design and completely open-source software, the AMiRo can be used for any application. Figure 1 shows a photograph of a covered and an uncovered version of the robot.

After a short description of the AMiRo's hardware and software structure in section 2, the demonstration of the system is described in section 3. It is split into three parts and includes multiple ways to remote control the robot, a fuzzy controller to follow lines, as well as a neural associative memory (NAM) implementing a Braitenberg vehicle.

2 Robot Architecture

The AMiRo features a modular design with a well defined physical and electrical interface. All software running on the microcontrollers (MCUs) or the SoC, as well as all applications and tools are open-source to ensure fast and easy development in any use case. Whilst the basic setup consists of three AMiRo modules (AMs), it can be easily extended with custom modules that satisfy the following reference design.

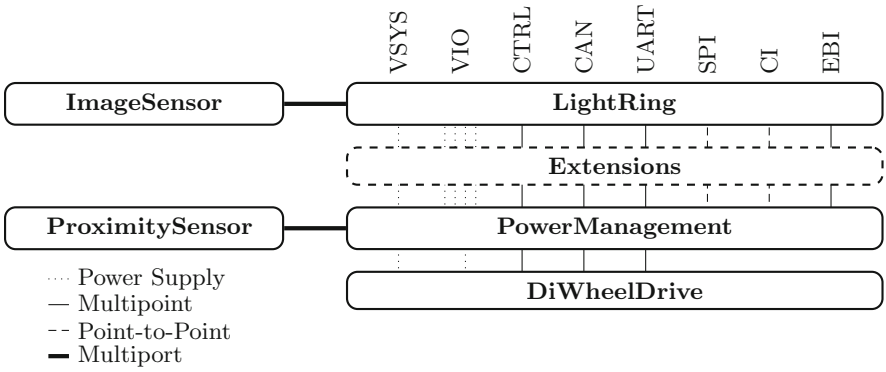


Fig. 2. Schematic of the system architecture including the electrical interfaces

2.1 Hardware

The AMiRo is shaped like a 100 mm wide cylinder with a height of 76 mm and moves on two wheels, which offers a differential kinematic. Its body is made of opaque acrylic glass and is divided into a lower tube-like part and a lid. Within the chassis, the modules are stacked upon each other and interconnect through a defined common interface. This interface consists of two 60 pin connectors that carry supply power, serial communication standards (CAN, UART, SPI),

a parallel camera interface (CI), an external bus interface (EBI), and dedicated control signals. Since CAN is used as the primary communication bus, each module must provide a controller and transceiver, whilst all other signals are optional. Following this reference, it is possible to design new AMs which add any required hardware to the AMiRo in order to make the system suitable for any application.

The basic setup is composed of three heterogeneous AMs. Starting from the bottom, the *DiWheelDrive* is dedicated to motion control, odometry, and floor detection. On top of the board, two U-profiles are mounted in order to create some space for the motors, wheels, and batteries. These are then roofed by the *PowerManagement*, which provides the system with energy, monitors the power consumption, and executes basic behaviors. Above, an arbitrary number of extension modules can be stacked to enhance the robots abilities. Finally, the *LightRing* terminates the module stack as the topmost AM just below the lid of the chassis. A schematic visualization of this architecture is given in figure 2.

Since the ***DiWheelDrive* AM** acts as bottom of the robot shell, two metal sliders are attached underneath for stabilization. An STM32F103 MCU from STMicroelectronics controls the two 1 W DC gear motors and collects information from the several sensors on the board. These sensors comprise optical encoders within the motors, a three axis gyroscope and accelerometer, a compass, and four infrared based proximity sensors with ambient light measurement facing the ground. For precise monitoring of the motors' power consumption, two measuring shunts are integrated as well. Furthermore, there are two charging pin pairs at the rear side of the board, which can be used for charging the batteries.

The ***PowerManagement* AM** hosts multiple sensors to monitor the status of the two 1950 mAh lithium-ion battery packs and the system's power consumption. Additional hardware is available to control charging and to generate four stable voltages besides the variable one from the batteries or an external power source. Such an external supply can be attached either to the charging pins of the *DiWheelDrive* or using the power connector on the *PowerManagement*. Additionally, the AM is extended by the *ProximitySensor*, which is a flexible PCB, integrated in the chassis. This sensor module hosts eight of the same proximity and ambient light sensors, as the *DiWheelDrive* plus four capacitive touch sensors. A flexible flat cable links the *ProximitySenor* with the *PowerManagement*. Since this AM is meant to perform basic behaviors, it is equipped with a powerful STM32F405 MCU from STMicroelectronics. In order to provide a low-power wireless communication interface, the *PowerManagemnet* also provides a BlueGiga WT12 Bluetooth chip.

The ***LightRing* AM** finalizes the module stack and is designed to fulfill three purposes. It features eight colored LEDs, that can be adjusted individually by the STM32F103 MCU, and a low-power radio module (A2500R24A Anaren Integrated Radio). Furthermore, two interfaces are provided to attach external devices. The first is a serial interface, which can be used to attach additional sensors, such as a Hokuyo URG LX04 laser range finder. The second is a 27

pin parallel CI, which can be connected to the *ImageSensor*, another small PCB integrated in the front of the robot's chassis. It accommodates a 5-megapixel camera (OmniVision OV5647) and is linked to the *LightRing* by a flexible flat cable. The CI signals are directly wired to the 60 pin connectors, so that any other AM can access the camera via the common communication interface.

Two **Extension modules** have been developed so far, which enhance the AMiRo's computing capabilities. The *Cognition* AM hosts a Gumstix Overo Tide Computer-on-Module (COM), which again features an OMAP3530 SoC from Texas Instruments and 512 MB RAM. An additional wireless module provides 802.11b/g/n capabilities and two speakers and microphones enable bidirectional sonic communication. The *ImageProcessing* AM features a Xilinx Spartan 6 FPGA, which is connected to all signals of the 60 pin connectors. Although it was primarily developed to accelerate image- and video processing, it can be used for any other application as well, such as artificial neural networks [7].

2.2 Software

The software habitat of the AMiRo is divided into three parts. First, the MCUs of the basic AMs run the OpenBLT bootloader and the AMiRo-OS, which configures the underlying ChibiOS/RT kernel and extends it by module specific hardware drivers. Several C and C++ interfaces provide access to all system functions in order to offer maximum flexibility for additional user applications. Second, the COM on the *Cognition* AM runs a custom Linux operating system. It is based on Poky [2], the reference distribution of the Linux Yocto Project [4]. Finally, multiple application programs for the MCUs and the COM, as well as tools, such as the Gazebo simulator, are provided. All mentioned software is open-source in order to involve the whole community in the further development.

3 Demonstration

The demonstration of the AMiRo showcases the sensory and computational capabilities of the basic setup as well as its easy extensibility. Starting with an embedded fuzzy control for line following, succeeded by a neural associative memory implementing a Braitenberg vehicle and several ways to remote control the AMiRo, all three basic AMs plus the *Cognition* are utilized. All tasks are shown on a table on which a small course is built up, the robot has to complete autonomously. Figure 3 shows an example for such a demonstration setup.

3.1 Line Following

This demonstration shows a fuzzy control running on the MCU of the *DiWheelDrive* and thus is tightly coupled to the motor controller. It utilizes the information of the proximity sensors on the bottom side of the board to follow lines by measuring the reflection coefficient.

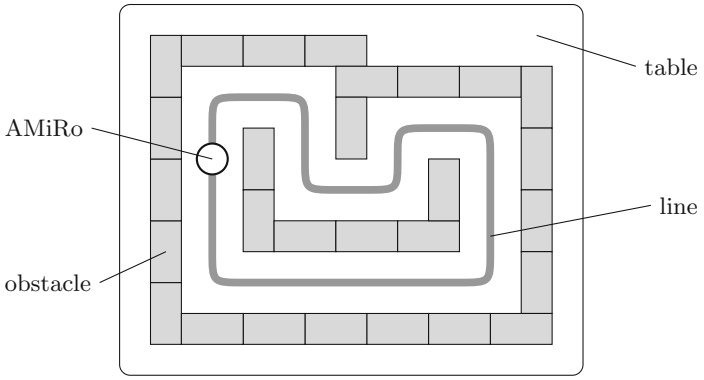


Fig. 3. Top view of an example setup for the demonstration

Since all computation of this behavior is executed by the MCU of the *DiWheeDrive* only, it has no impact on the performance of the rest of the system at all. As a result, exchanging or modifying the algorithm, so that the complexity strongly varies, has no influence on the behavior of any other AM. This is one example, how the decentralized architecture of the AMiRo allows to easily alter and evaluate single parts of the system.

3.2 Braitenberg Vehicle

The second demonstration showcases the whole basic setup with all three AMs involved. The measurements of the eight proximity sensors of the *Proximity-Sensor* module are used as input for a neural associative memory. This NAM is computed on the MCU of the *PowerManagement* and implements a Braitenberg vehicle. Thus, it generates outputs that can be applied to the motor controller of the *DiWheelDrive* and make the AMiRo avoid obstacles. Furthermore, the proximity inputs as well as the desired movement direction are displayed via the eight LEDs of the *LightRing*. The communication between the AMs is handled via CAN only.

Furthermore, the Braitenberg behavior can even be combined with the line following. As long as the *DiWheelDrive* does not detect any line at the ground, the output values of the NAM are used for motor control. Once a line is found, these commands are ignored and the robot follows that line until its end. Then, the Braitenberg behavior regains control over the motors until a line is encountered again.

3.3 Remote Control

Two ways to remote control the AMiRo are shown in this last demonstration. The first takes advantage of the Bluetooth capabilities of the basic setup and

a Wiimote of the Nintendo Wii game console. After simply pairing the robot with the controller, the user can navigate the AMiRo through the course by pressing buttons. Again, other behaviors like the Braitenberg vehicle may still be active, but get overwritten by the user input. Whenever no button is pressed, the AMiRo will fall back to the Braitenberg vehicle behavior automatically. This way it is possible to run multiple controllers simultaneously, which compete for execution by the *DiWheelDrive*.

Another way to control the AMiRo over any distance is an example application for the *Cognition* AM. In this demonstration it runs a WebSockets based webserver with a graphical user interface (GUI), that can be accessed via Wi-Fi. The GUI shows the current video stream from the front facing camera and several controls to steer the robot. This way it is possible to pilot the AMiRo over any distance via the web. Again, other behaviors can run simultaneously, so that the user may just watch the video stream while the robot is navigating autonomously.

Acknowledgments. Funded by the DFG, Excellence Cluster 277 "Cognition Interaction Technology".

References

1. Khepera IV. http://www.k-team.com/mobile-robotics-products/khepera-iv/ (accessed: April 04, 2015)
2. Poky linux. http://www.pokylinux.org/ (accessed: April 03, 2015)
3. Pr2: Robot for research and innovation. https://www.willowgarage.com/pages/pr2/overview (accessed: December 04, 2015)
4. Yocto project. https://www.yoctoproject.org/ (accessed: April 03, 2015)
5. Proceedings of the 1st international khepera workshop: Experiments with the mini-robot khepera (1999)
6. Herbrechtsmeier, S., Rückert, U., Sitte, J.: AMiRo – autonomous mini robot for research and education. In: Rueckert, U., Joaquin, S., Felix, W. (eds.) Advances in Autonomous Mini Robots. Non-series, vol. 101, pp. 101–112. Springer, Heidelberg (2012)
7. Lachmair, J., Merényi, E., Porrmann, M., Rückert, U.: A reconfigurable neuroprocessor for self-organizing feature maps. Neurocomputing **112**, 189–199 (2013)
8. Mondada, F., Franzi, E., Guignard, A.: The development of khepera. experiments with the mini-robot khepera. In: Proceedings of the First International Khepera Workshop, pp. 7–14. HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut (1999)
9. Rtornaz, P., Riedo, F., Magnenat, S., Vaussard, F.C., Bonani, M., Mondada, F.: Seamless multi-robot programming for the people: ASEBA and the wireless thymio II robot. In: 2013 IEEE International Conference on Information and Automation (ICIA), pp. 337–343. IEEE (2013)
10. Rückert, U., Joaquin, S., Felix, W.: Advances in autonomous mini robots. In: Proceedings of the 6th AMiRE Symposium. Springer Publishing Company, Incorporated (2012)

Video and Image Processing

Visualization of Complex Datasets with the Self-Organizing Spanning Tree

Ezequiel López-Rubio[1(✉)], Esteban José Palomo[1], Rafael Marcos Luque Baena[2], and Enrique Domínguez[1]

[1] Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain
{ezeqlr,ejpalomo,enriqued}@lcc.uma.es
[2] Department of Computer Systems and Telematics Engineering, University of Extremadura, University Centre of Mérida, 06800 Mérida, Spain
rmluque@unex.es
http://www.lcc.uma.es/∼ezeqlr/index-en.html

Abstract. Visualization of real world data is a difficult task due to the high-dimensional and the complex structure in real datasets. Scientific data visualization requires a variety of mathematical techniques to transform high-dimensional data sets into simple graphical objects that provide a clearer understanding. In this work a Self-Organizing Spanning Tree is proposed, which is able to learn a tree topology without any prespecified structure. Experimental results are provided to show the good performance with synthetic and real data. Moreover, the proposed self-organizing model is applied to color vector quantization, whose comparative results are provided.

Keywords: Self-organizing map topologies · Visualization · Spanning trees · Unsupervised learning

1 Introduction

Real world data are often very high-dimensional and usually have a complex structure that is difficult both to recognize and visualize. These complicated relations which exist in real datasets call for tools to visualize high-dimensional data, being crucial in pattern discovery. Scientific data visualization requires a variety of mathematical techniques to transform multivariate data sets into simple graphical objects that provide scientists and engineers with a clearer understanding of the underlying system behavior. Data analysis relies heavily on automated visualization tools to assist the user in discovering and analyzing patterns within large high-dimensional data sets.

Visualization involves mapping an unknown high-dimensional space onto a drawable structure. The role of visualization tools consist in discovering the relations among data vectors. Many approaches to data visualization focus either on extracting and representing the global topology of the input dataset, or

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 209–217, 2015.
DOI: 10.1007/978-3-319-19258-1_18

on extracting information about its clusters. Methods focused on dimensional reduction, such as self-organizing maps (SOM, [1,2]), do not represent cluster boundaries explicitly in a 2D structure, but they map high-dimensional data on a network of nodes so that vectors which are close in the input space are mapped onto nearby nodes of the network [3]. In this sense, the learned computational map structure preserves the topology of the input data set as much as possible [4].

Among dynamic topology self-organizing networks, tree structured models are very popular. A broad class of these models considers growing trees, so that more neurons and levels are added to the tree as required by a suitable data representation quality criterion [5]. This is useful when hierarchical data must be processed [6,7]. Other proposals consider a static model, i.e. the number of neurons does not change. This has some advantages, such as the limitation of the complexity of the model which makes possible to devote more effort to the correct adaptation of each unit so that its learned prototype represents a local region of the dataset in a better way. For example, the Self-Organizing Topological Tree [8] is a multilayer SOM with fixed depth and breadth, i.e. each neuron has a fixed number of children, and the topology among the children is the same for all parent units. The TreeSOM approach trains a standard SOM and then builds a hierarchical clustering [9]. Finally, a SOM can be trained along with user defined preferences about the connections among neurons, so that a tree topology is built which reflects those preferences [10].

Here our aim is to propose a new self-organizing network model with a fixed number of neurons which learns a tree topology among them with no prespecified structure. Hence it departs from known growing self-organizing trees, self-organizing trees with a fixed structure, and self-organizing trees with user defined structure.

The organization of the paper is as follows. The newly proposed Self-Organizing Spanning Tree (SOST) is described in the section 2. Experimental results are shown in the section 3. Finally, in the section 4 some conclusions and future works are presented.

2 The Model

In this section the Self-Organizing Spanning Tree (SOST) model is defined. This network considers a fixed number of neurons M. A dynamic topology is learned from the data, with the constraint that the links among neurons must form a tree, i.e. any two neurons must be connected by exactly one path. It must be noted that undirected links are considered here, i.e. given two neurons i and j, the links (i, j) and (j, i) are the same. The topological distance $d(i, i')$ between the neurons i and i' is defined as the length of the unique path which connects both neurons, given by the number of links. Figure 1 depicts an example of the SOST architecture.

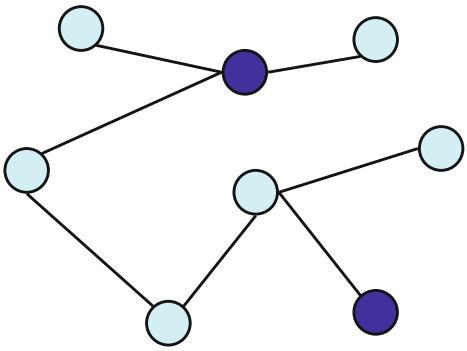


Fig. 1. Example of the SOST architecture. A network with $M = 8$ neurons is shown. The topological distance between the two neurons marked in a darker tone is $d(i, i') = 4$, because the path between i and i' contains 4 links.

Every neuron i has a prototype vector $\boldsymbol{\mu}_i$ which represents a cluster of input samples. Please note that $\boldsymbol{\mu}_i \in \mathbb{R}^D$, where D is the dimension of the input space. At time step n, a new sample $\mathbf{x}(n)$ is presented to the network, and a winner neuron is declared:

$$Winner(\mathbf{x}(n)) = \arg \min_{j \in \{1,...,M\}} \left\| \mathbf{x}(n) - \boldsymbol{\mu}_j(n) \right\| \tag{1}$$

Then the prototypes of all the units are adjusted, for $i \in \{1, ..., M\}$:

$$\boldsymbol{\mu}_i(n+1) = \\ \boldsymbol{\mu}_i(n) + \eta(n) \Lambda(i, Winner(\mathbf{x}(n)))(\mathbf{x}(n) - \boldsymbol{\mu}_i(n)) \tag{2}$$

where $\eta(n)$ is a decaying learning rate and the neighborhood function Λ varies with the time step n and depends on a decaying *neighborhood radius* $\Delta(n)$:

$$\eta(n+1) \leq \eta(n) \tag{3}$$

$$\Lambda(i, Winner(\mathbf{x}(n))) = \exp\left(-\left(\frac{d(i, Winner(\mathbf{x}(n)))}{\Delta(n)} \right)^2 \right) \tag{4}$$

$$\Delta(n+1) \leq \Delta(n) \tag{5}$$

At each time step, the second best matching neuron is also computed:

$$Second(\mathbf{x}(n)) = \arg \min_{j \in \{1,...,M\}, j \neq Winner(\mathbf{x}(n))} \left\| \mathbf{x}(n) - \boldsymbol{\mu}_j \right\| \tag{6}$$

Moreover, a counter of the number of times that a pair of neurons has been the first and second best matching ones is kept:

$$h_{ij}(n+1) = \begin{cases} h_{ij}(n) + 1 & \text{iff } (i = Winner(\mathbf{x}(n))) \wedge (j = Second(\mathbf{x}(n))) \\ h_{ij}(n) & \text{otherwise} \end{cases} \tag{7}$$

where $i, j \in \{1, ..., M\}$ so that the counters $h_{ij}(n)$ form a matrix $\mathbf{H}(n)$ of size $M \times M$ with zeros in the main diagonal.

The topology of the network is dynamically updated every certain fixed number of steps. To this end, the current value of the counter matrix $\mathbf{H}$ is used to compute a symmetric weight matrix $\mathbf{W}$ of size $M \times M$ with zeros in the main diagonal:

$$\mathbf{W} = \mathbf{H} + \mathbf{H}^T \tag{8}$$

Then the new topology is computed as the maximum spanning tree of the undirected graph whose nodes are the neurons and whose edge weights are given by $\mathbf{W}$. This is done so as to obtain the tree topology which connects all the neurons and has the maximum sum of weights. Hence the neurons which activate together more frequently are more likely to be connected by a direct link in the topology.

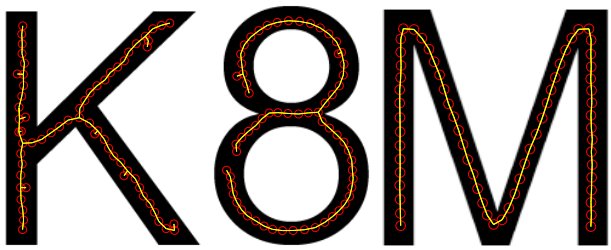
As usual in self-organizing neural networks the learning process is divided into two phases [1–3]. First an ordering phase is carried out where the neighborhood radius Δ and the learning rate η have a linear decay so that the network is able to capture the main features of the input dataset. After that, a convergence phase with constant values of Δ and η is executed, where a fine tuning of the network is attained.

3 Experimental Results

In this section, two different experiments have been carried out to show the performance of the SOST model. In Section 3.1, the self-organization capability of our proposal is demonstrated by using two-dimensional input distributions. Section 3.2 shows the image quantization performance of the SOST model.

3.1 Self-Organizing Experiments

This kind of experiments is devoted to show the unfolding capabilities of the SOST model when adapting its tree structure to the shape of two-dimensional input distributions with uniform density. For our experiments the 'K' letter, the '8' number and the 'M' letter input distributions have been selected. Each of these input distributions has $N = 10,000$ samples and two dimensions ($D = 2$). The training was done during 10 epochs and using 64 neurons. The resulting SOST networks are shown in Figure 2.



(a) (b) (c)

Fig. 2. SOST results for self-organization experiments: (a) two-dimensional 'K' letter input distribution, (b) two-dimensional '8' number input distribution, and (c) two-dimensional 'M' letter input distribution

In these plots, neurons are represented by red circles and connections among mean vectors of the neurons are plotted with straight yellow lines. By observing these plots, we can see how the tree structure of the SOST adapts correctly to the shape of the three input distributions.

3.2 Image Quantization Experiments

Several standard images have been analysed to prove the viability of this approach on image quantization. The selected images are: *Lena*, *Baboon*, *House* and *Peppers*, obtained from Matlab repository. The number of neurons of the SOST

Table 1. Performance of the SOST model on image quantization varying the number of neurons. The measures MSE (lower is better), PSNR and SSIM (higher is better) point out the quality of the compressed image.

Images	Measures	Num. Neurons						
		4	9	16	25	36	49	64
Lena	SSIM	0.7343	0.8332	0.8791	0.9134	0.9372	0.9438	0.9526
	PSNR	64.0637	68.3250	70.9656	72.6147	74.0163	75.0815	76.0265
	MSE	0.0257	0.0096	0.0052	0.0036	0.0026	0.0020	0.0016
Baboon	SSIM	0.6412	0.7913	0.842	0.8808	0.9021	0.9207	0.9282
	PSNR	60.8351	64.3821	66.1869	67.973	69.1698	70.1131	70.7786
	MSE	0.0541	0.0239	0.0158	0.0105	0.0079	0.0064	0.0055
House	SSIM	0.8064	0.8447	0.919	0.9479	0.9588	0.9701	0.9769
	PSNR	65.1698	69.1983	71.3485	73.1198	74.6834	75.6047	76.2098
	MSE	0.0199	0.0079	0.0048	0.0032	0.0022	0.0018	0.0016
Peppers	SSIM	0.6906	0.7559	0.839	0.8804	0.9059	0.9079	0.9273
	PSNR	61.4159	65.2174	68.0882	69.7431	70.7304	71.6157	72.5393
	MSE	0.0473	0.0197	0.0102	0.007	0.0055	0.0045	0.0037

model has varied from one configuration to another, where this parameter can take the following values: 4, 9, 16, 25, 36, 49 and 64. It should be noted that the number of prototypes represents the number of patterns used to color the image. Several common measures related to the image quality have also been considered: the mean square error (MSE), the peak signal noise ratio (PSNR) and the structural similarity index (SSIM) [11]. The former measure is computed using the code included in [12], over the gray image converted from the rgb one.

Table 1 shows a comparative of the SOST approach with several configurations, varying the number of neurons. It is possible to observe that the quality of the compressed image with the SOST model is suitable from 16 neurons (upper than 0.80 in SSIM), specially in the images *Lena* and *House*, where the tonality of the whole images is more homogeneuos.

(a) (b) (c) (d)

Fig. 3. SOST results for *Lena* image: (a) Original and compressed with (b) 4 neurons, (c) 16 neurons and (d) 64 neurons

(a) (b) (c) (d)

Fig. 4. SOST results for *Peppers* image: (a) Original and compressed with (b) 4 neurons, (c) 16 neurons and (d) 64 neurons

Figures 3 and 4 show the quantized images obtained by the SOST model with 4, 16 and 64 neurons. The last image with 64 neurons achieves a great similarity with the original one.

In order to analyze the differences in performance with other self-organizing maps, our SOST approach is compared with the 2D self organizing map

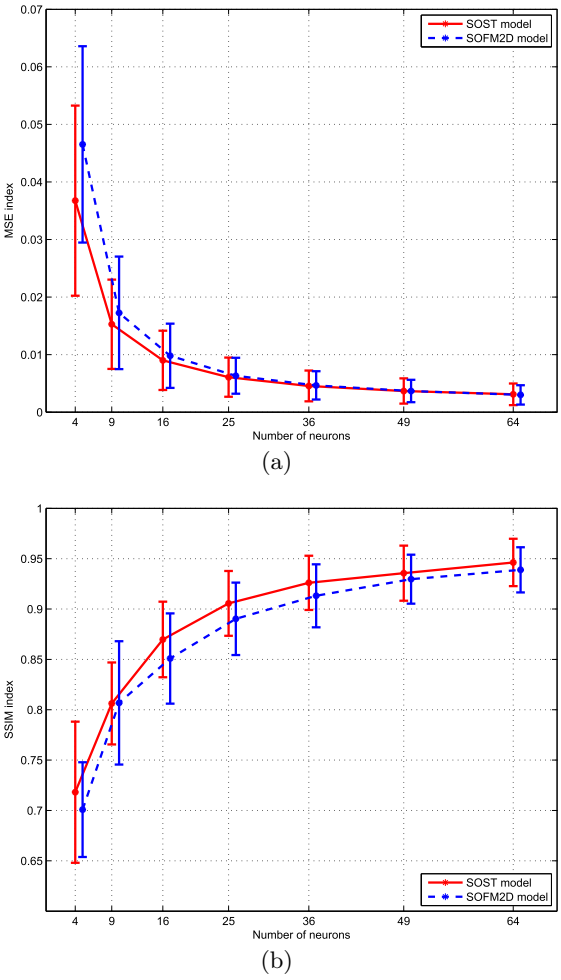


Fig. 5. Comparative of our SOST approach and the SOFM2D model for all the images analysed by using two metrics (mean and standard deviation): (a) MSE and (b) SSIM

(SOFM2D) [1], where the number of neurons is the same for both models. The topology map for the SOFM2D is a square of $\sqrt{number\ of\ neurons}$ for each side. Figure 5 shows this comparative. It is possible to observe that, in general, the results from the SOST model are quite competitive, even better with different number of neurons than the obtained from the SOFM2D model, according to the both metrics (MSE and SSIM). Additionally, Figure 6 shows a visual comparative of the compression of the images with the two neural models (SOST and SOFM2D). It is highlighted the graduation of the color of SOST model, improving the visual quality of the image in comparison with the results of the SOFM2D alternative.

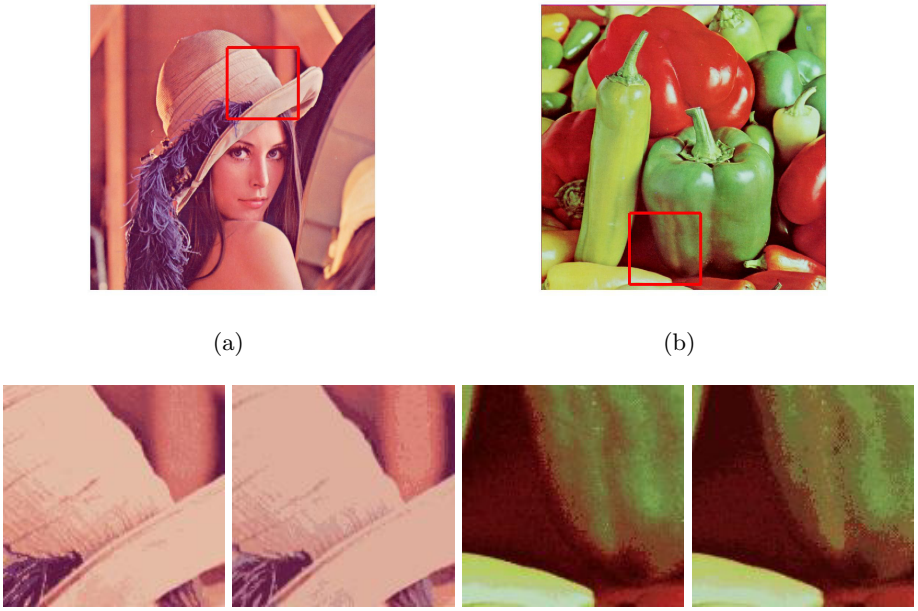


Fig. 6. Qualitative comparison between SOST and SOFM2D model. (a) and (b) Original images; (c) and (e) SOST results; (d) and (f) SOFM2D results.

4 Conclusions

A new self-organizing network model has been proposed. It considers a fixed number of neurons, and it learns a tree topology among them from the input dataset, so that the connectivity of the input is adequately modeled. Hence it departs from previously proposed tree SOM approaches, which do not allow a dynamic rearrangement of the connections among units. The algebraic theory of spanning trees is used to learn the topology, so that the network represents the input structure in the best possible way. Experimental results have been presented both with synthetic and real data, with an application to color vector quantization. This demonstrates the usefulness of our proposal for real world tasks.

Acknowledgments. This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grant TIN2011-24141, project name Detection of anomalous activities in video sequences by self-organizing neural systems. It is also partially supported by the Autonomous Government of Andalusia (Spain) under projects TIC-6213, project name Development of Self-Organizing Neural Networks for Information Technologies; and TIC-657, project name Self-organizing systems and robust estimators for video surveillance. Finally, it is partially supported by the Autonomous Government of Extremadura (Spain) under the project IB13113. All of them include

funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga.

References

1. Kohonen, T.: Self-Organizing Maps, 3nd edn. Springer (2001)
2. Kohonen, T.: Essentials of the self-organizing map. Neural Networks **37**, 52–65 (2013)
3. Yin, H.: The self-organizing maps: Background, theories, extensions and applications. Studies in Computational Intelligence **115**, 715–762 (2008)
4. Astudillo, C., Oommen, B.: Topology-oriented self-organizing maps: A survey. Pattern Analysis and Applications **17**(2), 223–248 (2014)
5. Pakkanen, J., Iivarinen, J., Oja, E.: The evolving tree - analysis and applications. IEEE Transactions on Neural Networks **17**(3), 591–603 (2006)
6. Luo, F., Khan, L., Bastani, F., Yen, I.L., Zhou, J.: A dynamically growing self-organizing tree (dgsot) for hierarchical clustering gene expression profiles. Bioinformatics **20**(16), 2605–2617 (2004)
7. Doan, N.Q., Azzag, H., Lebbah, M.: Growing self-organizing trees for autonomous hierarchical clustering. Neural Networks **41**, 85–95 (2013)
8. Xu, P., Chang, C.H., Paplinski, A.: Self-organizing topological tree for online vector quantization and data clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **35**(3), 515–526 (2005)
9. Samsonova, E., Kok, J., IJzerman, A.: Treesom: Cluster analysis in the self-organizing map. Neural Networks **19**(6–7), 935–949 (2006)
10. Astudillo, C., John Oommen, B.: Imposing tree-based topologies onto self organizing maps. Information Sciences **181**(18), 3798–3815 (2011)
11. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
12. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: The SSIM Index for Image Quality Assessment (2003). https://ece.uwaterloo.ca/z70wang/research/ssim/ (accessed January 31, 2015)

A Detection System for Vertical Slot Fishways Using Laser Technology and Computer Vision Techniques

Angel J. Rico-Diaz[1,2] (✉), Alvaro Rodriguez[1], Daniel Villares[2], Juan R. Rabuñal[2], Jeronimo Puertas[3], and Luis Pena[3]

[1] Dept. of Information and Communication Technologies, University of A Coruña. Campus de Elviña, 15071 A Coruña, Spain
{angel.rico,arodriguezta, dvillares,juanra,jpuertas,lpena}@udc.es
[2] Innovations in Construction and Civil Engineering (CITEEC), University of A Coruña. Campus de Elviña, 15071 A Coruña, Spain
[3] Dept. of Hydraulic Engineering, ETSECCP, University of A Coruña. Campus de Elviña, 15071 A Coruña, Spain

Abstract. Vertical slot fishways are structures that are placed in rivers to allow fish to avoid obstacles such as dams, hydroelectric plants. Knowing the frequency with which fish go through this type of structures can help to determine their efficiency, as well as know migratory features from species, determine if the fluvial course is healthy or if it is possible to fish with fauna preservation guarantees.

A non-invasive method for fish detection, without the need of direct observation, which uses a laser sensor and computer vision techniques to detect fish, is proposed in this work.

Keywords: Computer vision · Fishways · Fish counter · Laser detection

1 Introduction

Vertical slot fishways are hydraulic structures that allow fish movement upwards in rivers with obstacles such as dams, hydroelectric plants and others.

Along the years, several researches have been conducted to study fishways and fish, including water flow features [1-3], fish swimming abilities [4, 5] or fish behavior within them [6]. In this work it is intended to determine in an easy way, in real time and with reduced cost the specimen number that go through the fishway, since knowing this value will serve to estimate the structure effectiveness [7], determine the fluvial course health, or river fish use possibilities.

One of the techniques that is used for fish detection is the sonar. The sonar is a device that detects submerged objects presence and situation through acoustic waves. From the sixties this technique has been used in applications to detect fish [8] or study fish features [9]. More recently, systems as the DIDSON (Dual-frequency identification sonar), have reduced the previous acoustic systems limitations, allowing the

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 218–226, 2015.
DOI: 10.1007/978-3-319-19258-1_19

obtaining of higher quality images, and favoring the development of new studies for fish detection and counting [10-13].

In other works techniques based on infrared laser have been used for fish detection [14]. One of the most well-known fish counter, the *Riverwatcher Fish Counter* [15], is based on this technique.

Finally, other authors have obtained promising results have been obtained using submarine cameras combined with computer vision techniques [16, 17] or Artificial Neural Networks (ANN) [18].

The general process of a fish detection application based on Computer Vision, once the image has been captured, is usually divided into a series of phases. During the first stages it is attempted to improve the image quality to be able to obtain useful information for subsequent stages.

In the proposed technique it is combined, on the one hand, the use of an infrared laser sensor to make the fish detection and, on the other hand, computer vision techniques to prove that the object detected by the laser is really a fish.

In this way, a fish detection system in real time is obtained, reducing the computational time of Artificial Neural Networks, and at a reduced cost, since an expensive assembly is not required.

2 Proposed Technique

In this work a technique to detect and monitor the fish density that goes through a vertical slot fishway, with low cost and in real time, is proposed. To this end, computational and Computer Vision techniques are used to analyze the data obtained with a laser sensor and a camera.

Therefore, a system with different components has been designed. It uses two kinds of sensors: a conventional camera and a laser sensor, focused on the same fishway section. Therefore, the camera captures images from the objects detected when interfering with the beam emitted by the laser sensor (Fig. 1).

Fig. 1. Proposed system

This information is downloaded in a storage unit where the computer vision software application is executed. In addition, a control and a display unit are employed to allow the interaction of the user with the application.

The technique can be divided into two stages that are described in detail as follows:

2.1 Object Detection Stage with the Laser Sensor

The sensor is placed in a fishway section of interest to study the crossing fish. Laser measurements are encoded as grey scale images (Fig. 2a), where the grey level in a point represents the measured distance by the laser in that point. These values are stored in a buffer, in a way that each time the sensor sends data, they are introduced at the end of the buffer and the rest of the vectors are moved (Fig. 3). In this way it is obtained a distance image from the laser influence area that is refreshed in real time.

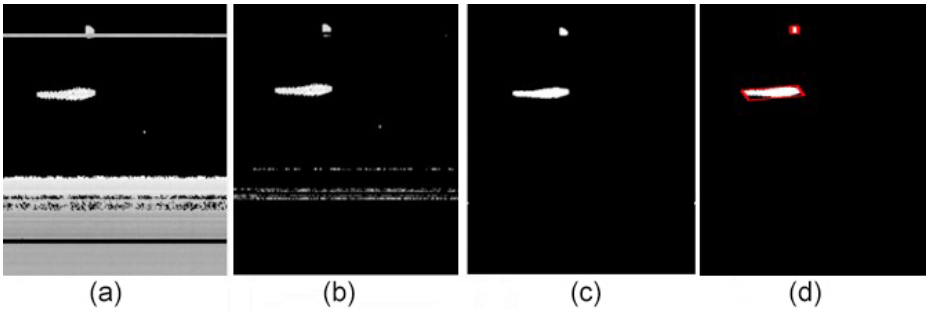


Fig. 2. Applied computer vision process. (a) Generated image, (b) image after the pre-processing, (c) image after the segmentation process (d) image with the selected candidate objects.

After the image is obtained, the computer vision process is executed, it includes the following stages: pre-processing, segmentation and recognition and interpretation.

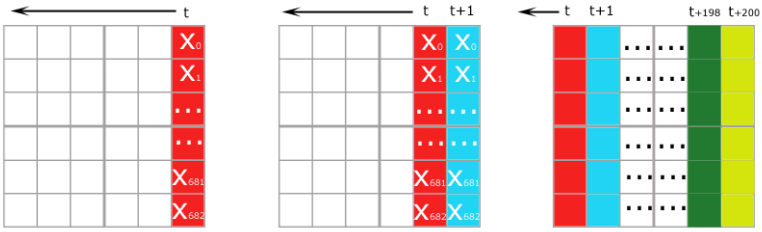


Fig. 3. Functioning of the sensor reading value input in the buffer

The image pre-processing is focused in improving the image quality for future treatment and to facilitate the image analysis in subsequent stages. In this first stage the image background is eliminated using a smoothed sample scan of the fishway. This sample is subtracted to each data vector that the sensor sends. The canvas where the image is then represented is defined as black for subsequent stages (Fig. 2b).

The next step of the process is a segmentation algorithm. Segmentation is the process of dividing an image in the parts that constitute it or the objects that form it. In this way, objects of interest can be separated from the ones that are non-relevant, which are considered as background. This stage is one of the basic stages in any computer vision automatic system, both for the difficulties that entails and for the importance of its results.

Here, the binary thresholding technique, that allows making the segmentation in real time, has been used.

With the aim of improving the results obtained in the previous step, morphological operations are applied [24] to the image in the next step. Mathematical morphology is a theoretical framework for non-linear processing of digital images. With this technique, it is possible to quantify and filter specific geometric structures contained in an image [25]. In this case opening and closing operations have been used [19, 20] to improve the detected object edges. The opening operation allows filtering the smallest objects and the closing operation fills the gaps that appear in the remaining objects of interest. In this way, a combination of one opening operation followed by three closing operations are used to remove noise and optimize the result (Fig. 2c).

After the segmentation phase, a representation and description stage is executed on the objects that have been previously segmented. The aim at this technique is choosing a useful computational representation of selected features of the objects, which remain invariant to geometric transformations.

This representation is made through the minimum rectangle that contains the object, allowing a second filtering stage, where each object is selected for the next stage, or dismissed according to its size (Fig. 2d).

2.2 Verification- Rejection Stage

In the previous stage an object has been detected in the water and it has been marked as a candidate to be a fish. To verify its condition, a snapshot of the object is taken and analyzed. This analysis can be divided in a pre-processing stage, a segmentation stage and a representation and interpretation stage.

The pre-processing stage consists in a background subtraction process using a dynamic background modelling technique. According to this, every pixel of the scene must be matched to the background or foreground category. To this end a widely used model based in estimate the RGB color space probability distribution for every pixel in the image has been chosen.

The segmentation algorithm works using Bayesian inference to calculate the likelihood of a pixel $x_{ij}(t)$, observed at time t in coordinates (i,j), being classified as background (B) or foreground (F). This can be expressed as follows:

$$p(B \mid x) = \frac{p(x \mid \mathrm{B})\, p(B)}{p(x \mid \mathrm{B})\, p(B) + p(x \mid F)\, p(F)} \qquad (1)$$

$$p(x \mid \mathrm{B}) \sim x_{ij}(t)^T\, H_{ij}(t)$$

Where $H_{ij}(t)$ is an histogram estimation in time t for the coordinates (i,j). This estimation is based in a set of observations $\chi_{ij} = \{x_{ij}(t-T), \ldots, x_{ij}(t)\}$ where T is a time period used to adapt to changes. According to it, new samples are added to the set and old ones are discarded, while the set size does not exceed a certain value. According to this, the estimated probability distribution will be adapted to changes in the scene. This is performed by using weighting values depending a on decaying factor that is used to limit the influence of the old data.

These kind of algorithms have been extensively studied in several works, being mostly based in the use of parametric models to estimate the probability distribution of pixel values [19, 20]. These techniques are based commonly on the use of single Gaussian distribution models or mixture of Gaussian models, according to the complexity of the scene [21, 22].

In this work, a nonparametric algorithm proposed in [23] has been used. Therefore, the distribution itself is estimated and updated. Taking this into account, and assuming that *a priori* probabilities $p(B)=p(F)=0.5$, a pixel is classified as a background if $p(x|B)>0.8$. The weights of the model are updated according to the following equation:

$$H_{ij}(t+1) = (1-\alpha) H_{ij}(t) + \alpha x_{ij}(t) \tag{2}$$

Where $x_{ij}(t)$ is the current sample and $\propto$ is a constant value, equivalent to $1/T$, as expressed above. In this work, $\propto$ was set by default to 0.025. Given that, the new weighting value in the histogram for a sample y observed k frames in the past and which had a weight w, will be $w(1-\propto)^k$.

The result of the background subtraction process is shown in Fig. 4b. At this point, a thresholding will be performed on the result, obtaining a binary representation of the candidate foreground objects. Finally a representation and interpretation phase is conducted, using the edges of the detected objects, whose aim is determining if they can be considered as fish (Fig. 4c). To this end, the detected objects will be compared with a data base of models that represent a fish shape.

Fig. 4. Applied computer vision process in verification-rejection stage. (a) Original image (b) Foreground objects after background subtraction (c) Fish representation. Color has been modified for visualization.

Firstly the segmented objects that do not achieve certain restrictions are eliminated, these restrictions are defined as valid ranges on the object size or its area, defined according to the fish species. In the remaining objects, the seven Hu moments [26] are calculated and used to compare the shape of the object with the fish models. Spatial moments [27] are statistical measurements that describe regions (objects) in terms of its interior points. The Hu moments are the normalized moments that are invariant under translation, changes in scale and under rotation.

The similarity between the compared figure and the segmented object is compared using the similarity value defined as follows:

$$f\left(A,B\right)=\max_{i=1:7}\frac{\left|m_i^A-m_i^B\right|}{\left|m_i^A\right|} \tag{3}$$

Where A is the segmented object and B is a model of a fish in the database, being m_i^A the i moment of Hu calculated in the object A.

The segmented object A is accepted as a fish (fish counter is increased) if the similarity value with any of the fish models in the database reach a threshold, set by default to 0.31; otherwise it is discarded.

3 Results

Two types of experimental tests have been conducted. In the first experiment, the laser sensor behavior in the water has been measured. The second type of tests is focused in evaluating the performance of the computer vision software.

3.1 Laser Performance

Two types of hydraulic channels have been used for this test: a 93 liters swimming pool and a circular channel with an engine that moves the water in a certain speed in a closed circuit. The tests have been made firstly with clean water to measure the laser range and afterwards it has been added turbidity (through the incorporation of slime silica) to analyze the relation of water turbidity with laser range.

With these measurements it could be analyzed how the sensor behaves in different circumstances, comparing the measurement values from the sensor with the real position of the fish, allowing to establish a relationship between the variables.

Four turbidity levels have been defined. The first level represent clean water and at each level the turbidity is increased till reaching level four, where there is not enough visibility to see objects with direct observation.

During the experiments it has been confirmed that the laser behavior in the water makes the sensor measurement higher than the real distance of the object. Additionally, the sensor loses accuracy when the distance of the object exceeds a certain limit. This aspect has been taken into account to calibrate the system to obtain a distance estimation according to reality.

It has also been confirmed that the materials that behave better are glass and quartz glass, since they allow measuring farther correct distances than the measurements that methacrylate offers.

3.2 Software Analysis

These tests have been conducted on a 1:8.3 scale model of a vertical slot fishway, built in methacrylate and connecting two fish tanks, where around ten fish of different colors and types were introduced to evaluate the implemented software efficiency.

Different tests have been conducted in this scenario. Each one of them lasted from 1 to 3 hours to produce enough fish crossings along the fishway to be significant.

Additionally, different non fish bodies, have been introduced in the assay, ranging from wood pieces to methacrylate and steel bodies, in order to determine the rate of false positives of the system.

Finally, to measure the behavior of the system in these assays, the obtained images of the detected objects have been stored and analyzed by experts. Around 200 images of crossing objects were analyzed (Table 1). The images where the complete object cannot be seen or the object is out of the shot have been discarded to obtain a reliable measure of the systems performance.

The metrics used to analyses the behavior of the system are *Precision* and *Recall*, shown in the following equations (Eq.4-7):

$$\text{Precision} = ((\text{True Positives}) / (\text{True Positives} + \text{False Positives})) \quad (4)$$

$$\text{Recall} = ((\text{True Positives}) / (\text{True Positives} + \text{False Negatives})) \quad (5)$$

$$\text{False Positive Ratio} = 1 - \text{Precision} \quad (6)$$

$$\text{False Negative Ratio} = 1 - \text{Recall} \quad (7)$$

Table 1. Test results

Fish	Data	%
Precision	0.97	97.18%
Recall	0.73	72.63%
False positive rate	0.03	2.18%
False negative rate	0.27	27.36%

4 Conclusions and Future Work

Analyzing the obtained results in the analysis of the different materials in which the laser was submerged, it could be concluded that it should be used quartz glass or glass to operate, since they are the ones that allow accurate measures with a higher range with the sensor beam. In addition it must be taken into account the refraction effect in water, which causes the distances to be overestimated by the device.

Analyzing the obtained results (Table1), a very high accuracy has been obtained, indicating that the system is able to detect the 73% of the fishes, with a 0.03% rate of wrong objects marked as fish.

It has been developed a fish detection and counter system using laser technology and computer vision techniques. It has been implemented in two phases: the first one, where the laser detects the crossing objects and a second one, where the detected objects are analyzed, studying their visual to confirm if they are fish.

This system offers a non-invasive method, working in real time, and without the visual limitation problem of direct observation techniques.

The main advantages of this technique are the following. The assembly size of the entire system is reduced, and the only required devices are: a laser sensor, a camera and a laptop with the software that makes the computer vision process.

In conclusion, the proposed technique can be carried out with low cost, and allows to sample a fishway section in real time with high precision.

The proposed method has been tested in a laboratory, simulating unfavorable conditions and obtaining promising results. Nevertheless, in next stages of this research, the system should be optimized and tested in a real scenario and in a real river. It may provide a useful fish counter tool to detect and study fish in their natural environment.

Acknowledgments. This work was supported by FEDER funds and Spanish Ministry of Economy and Competitiveness (Ministerio de Economía y Competitividad) (Ref. CGL2012-34688). The authors would also like to thank the Spanish Ministry of Education (FPI grant Ref. BES-2013-063444)

References

1. Wu, S., Rajaratma, N., Katopodis, C.: Structure of flow in vertical slot fishways. Journal of Hydraulic Engineering **125**, 351–360 (1999)
2. Puertas, J., Pena, L., Teijeiro, T.: An Experimental Approach to the Hydraulics of Vertical Slot Fishways. Journal of Hydraulics Engineering **130** (2004)
3. Tarrade, L., Texier, A., David, L., Larinier, M.: Topologies and measurements of turbulent flow in vertical slot fishways. Hydrobiologia **609**, 177–188 (2008)
4. Dewar, H., Graham, J.: Studies of tropical tuna swimming performance in a large water tunnel– Energetics. Journal of Experimental Biology **192**, 13–31 (1994)
5. Blake, R.W.: Fish functional design and swimming performance. Journal of fish biology **65**, 1193–1222 (2004)
6. Rodriguez, A., Bermudez, M., Rabuñal, J., Puertas, J., Dorado, J., Balairon, L.: Optical Fish Trajectory Measurement in Fishways through Computer Vision and Artificial Neural Networks. Journal of Computing in Civil Engineering **25**, 291–301 (2011)
7. Puertas, J., Cea, L., Bermudez, M., Pena, L., Rodriguez, A., Rabuñal, J., et al.: Computer application for the analysis and design of vertical slot fishways in accordance with the requirements of the target species. Ecological Engineering **48**, 51–60 (2012)
8. Craig, R.E., Forbes, S.T.: Design of a sonar for fish counting (1969)
9. Ehrenberg, J.E.: A method for extracting the fish target strength distribution from acoustic echoes. In: Ocean 72-IEEE International Conference on Engineering in the Ocean Environment, pp. 61–64 (1972)

10. Balk, H., Lindem, T.: Improved fish detection in data from split-beam sonar. Aquatic Living Resources **13**, 297–303 (2000)
11. Belcher, E., Matsuyama, B., Trimble, G.: Object identification with acoustic lenses. In: MTS/IEEE Conference and Exhibition OCEANS, pp. 6–11 (2001)
12. Han, J., Honda, N., Asada, A., Shibata, K.: Automated acoustic method for counting and sizing farmed fish during transfer using DIDSON. Fisheries Science **75**, 1359–1367 (2009)
13. Holmes, J.A., Cronkite, G.M., Enzenhofer, H.J., Mulligan, T.J.: Accuracy and precision of fish-count data from a "dual-frequency identification sonar"(DIDSON) imaging system. ICES Journal of Marine Science: Journal Du Conseil **63**, 543–555 (2006)
14. Mitra, V., Wang, C.-J., Banerjee, S.: Lidar detection of underwater objects using a neuro-SVM-based architecture. IEEE Transactions on Neural Networks **17**, 717–731 (2006)
15. Baumgartner, L., Bettanin, M., McPherson, J., Jones, M., Zampatti, B., Beyer, K.: Assessment of an infrared fish counter (Vaki Riverwatcher) to quantify fish migrations in the Murray-Darling Basin. Industry & Investment NSW, Fisheries Final Report Series **116**, 47 (2010)
16. White, D., Svellingen, C., Strachan, N.: Automated measurement of species and length of fish by computer vision. Fisheries Research **80**, 203–210 (2006)
17. Zion, B., Alchanatis, V., Ostrovsky, V., Barki, A., Karplus, I.: Real-time underwater sorting of edible fish species. Computers and Electronics in Agriculture **56**, 34–45 (2007)
18. Storbeck, F., Daan, B.: Fish species recognition using computer vision and a neural network. Fisheries Research **51**, 11–15 (2001)
19. Coifman, B., Beymer, D., McLauchlan, P., Malik, J.: A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies **6**, 271–288 (1998)
20. Horprasert, T., Harwood, D., Davis, L.S.: A statistical approach for real-time robust background subtraction and shadow detection. In: IEEE ICCV, pp. 1–19 (1999)
21. KaewTraKulPong, P., Bowden, R., An improved adaptive background mixture model for real-time tracking with shadow detection. In: Video-based surveillance systems, pp. 135–144. Springer (2002)
22. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, pp. 28–31 (2004)
23. Godbehere, A.B., Matsukawa, A., Goldberg, K.: Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. American Control Conference (ACC) **2012**, 4305–4312 (2012)
24. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image analysis using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 532–550 (1987)
25. Morales R.R., Azuela, J.H.S., Procesamiento y análisis digital de imágenes. Ra-Ma (2011)
26. Ochoa Somuanom, J., Pérez Lara, C., Toscano Martínez, J.H., Pereyra Ramos, C.G.: Clasificación de objetos rígidos a partir de imágenes digitales empleando los momentos invariantes de Hu. Presented at the X Congreso Internacional Sobre Innovación y Desarrollo Tecnológico, Cuernavaca Morelos, México (2013)
27. Pajares Martinsanz, G., De la Cruz García, J.: Visión por computador imágenes digitales y aplicaciones, 2ª edn. Ra-Ma (2007)

Interactive Relevance Visual Learning for Image Retrieval

Hsin-Chia Fu[1(✉)], Z.H. Wang[1], W.J. Wang[1], and Hsiao-Tien Pao[2]

[1] College of Engineering, Huaqiao University, Quanzhou, Fujian, China
hcfu99@gmail.com, {15980081901,18250621652}@qq.com
[2] Department of Management Science,
National Chiao-Tung University, Hsinchu, Taiwan, ROC
htpao123@gmail.com

Abstract. This paper proposes mixture Gaussian neural networks (MGNN) to learn visual features from user specified query image objects or regions for relevance image retrieval. Instead of segmenting query image regions from sample images, relevance feedback feature learning is performed by the proposed MGNN to extract query visual features. After feature learning, the MGNN can be used to measure the appearance difference between the query features and images for image retrieval. The proposed methods were tested on COREL image gallery and the WWW image collections, and testing results were compared with currently leading approaches. From the experimental results, that the extracted and learned query visual features by MGNN can be very close to users' mind and/or desire, and the closeness is somewhat related to the number of feature leaning iterations. Since any dimensional data can be approximated by mixture Gaussian distributions, thus using MGNN to query and to retrieve similar and/or relevance high dimensional data or images will be a new area of research for future works.

Keywords: Content-based image retrieval · Visual keywords · Mixture gaussian distribution · Reinforced and anti-reinforced learning · Decision-based neural network

1 Introduction

The continuous spread of digital media available over Internet has spawned great demand for searching of desired visual contents in large image database or WWW collections [1]. For years, text-based image search techniques have been used to retrieve images with text annotations. Apparently, text-based search techniques may cause visual annotation inaccuracy due to subjectivity of human perception [2, 3]. To overcome the disadvantages in text-based retrieval approaches, content-based image retrieval (CBIR) was introduced in the early 1980s. A pioneering work was published by Chang in 1984 [4] by using picture indexing and abstraction for pictorial database retrieval. Then, a few commercial products and experimental prototype systems have been developed, such as QBIC [5], Photobook [6], Virage [7], VisualSEEK [8],

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 227–240, 2015.
DOI: 10.1007/978-3-319-19258-1_20

Netra [9], SIMPLIcity [10] in the late 1990s. Some CBIR systems applied global features, such as color histograms or transform coefficient histograms. However, using global features for image indexing, a query system may include more features than the needed significant local details, so as to retrieve undesired results [11, 12].

Instead of using global features of an image, some later query systems adopted local features to represent or to index one or more regions in an image. As it has been found that users are usually more interested in specific regions rather than the entire image, thus representation of images at region level is proved to be more close to human perception system [13]. In fact, the performance of region-based image retrieval (RBIR) depends heavily on the accuracy of the segmentation of feature regions, which has been believed to be difficult off-line processing tasks [14]. Hence, on-line processing algorithms, such as relevance feedback (RF) [15] was proposed to learn the users' intentions on the fly. However, sometimes the identified visual features may still not fit to users' desire. In this paper, we focus on the interactive relevance visual feature learning for RBIR.

Mixture Gaussian neural networks (*MGNN*) [16] is processed to learn visual features from user specified objects or regions for relevance query and retrieval. Instead of using off-line segmenting query regions from sample images, interactive relevance feature learning is performed between users and *MGNN* to learn desired query visual features. After feature learning, *MGNN* measures visual difference between the query regions and candidate images to retrieve desired images. Also, a visual relevance image query (VRIQ) prototype was implemented to test and to evaluate the performance of the proposed approach.

The rest of paper is organized as follows. In Section 2, *visual features* and MGNN learning schemes are presented. Section 3 describes the proposed visual relevance image query approach. Experimental results are provided and discussed in Section 4. Finally, conclusions are reported in Section 5.

2 Visual Feature Learning

Similar to the keywords used to searching relevant documents over WWW, visual keywords [17, 18] are user specified image regions which can be used as visual features for RBIR. Usually, RBIR needs to precisely segment query regions from sample images to define user expected query concepts and then to generate semantic template for high-level image retrieval. However, precise off-line region segmentation is not trivial, and it may not be a necessary procedure to obtaining sematic templates. Instead, we propose an interactive relevance visual feature learning between user specified image regions and mixture Gaussian neural networks (MGNN) [cf. Sec II.B].

2.1 Interactive Relevance Visual Feature Learning

Figure 1 depicts the proposed interactive relevance feature learning procedure, which demonstrates how to extract proper visual features from user desired regions. The sample image is first partitioned into equal sized small blocks. Suppose the designated query region is the white horse, the user may click a mouse pointer on a representative block at the body of the white horse, as shown in Figure 1 (a). The color and

texture features of the selected block are collected and converted to mixture of Gaussian distributions [cf. Eq. (2)], which are used as the initial parameter values of MGNN (cf. Sec. 2.1.3). Then, all the partitioned small blocks of the sample image are input to MGNN for the measuring of visual similarity with respect to the user designated block. If the similarity value exceeds a predetermined threshold, the corresponding block in the sample image will be check marked. When all the blocks in the sample image are evaluated for visual similarity check with the selected block, their similarity status may be categorized into the following three cases [cf. Fig. 1 (b)]:

Case 1: Blocks in designated regions with a green check mark indicate that their visual features were properly matched with the selected block, and the MGNN has learned the visual features of these matched blocks;

Case 2: Blocks in designated regions with no check mark indicate that the MGNN needs to be reinforced learning [cf. Eq. (3)] by using the visual features in these blocks;

Case 3: Blocks outside of designated regions with red check marks indicate that the MGNN needs to be anti-reinforced learning [cf. Eq. (4)] by using the visual features in these blocks.

For the erroneous check marks in Cases 2 and 3, the MGNN usually needs to go through several iterations of reinforced and anti-reinforced learning by using their corresponding blocks in Cases 2 and 3 respectively, until all the blocks in the designed regions are checked [cf. Figure 1 (d)], and all the blocks outside of designated regions without check marks [cf. Figure 1 (c)]. As shown in Figure 1 (d), there are some green marks at the beach spray region, which were mistakenly checked marks. Thus, additional anti-reinforcement learning is needed to unlearn the light brown visual features from the MGNN.

2.2 Mixture Gaussian Neural Networks

Given an image I, a region A_i in I is used as a sematic template for region based image query and retrieval. A visual keyword ω_i is defined as mixture of Gaussian distributions to formulate the spatial，color, and texture features of the region A_i. Approximating a region A_i by mixture of Gaussian distributions has the following two b enefits: (1) to avoid complicate and difficult image segmentation processes, and (2) to allow flexibility in matching with specified visual features [19]. In the followings, how to define visual keywords for an image region will be introduced first.

As noted previously, a region can be represented by its, color, texture and spatial features. Since the Color Measurement Committee (the CMC) distance [20] in the CIELab color space is one of most accurate metrics according to human perception, we employ the CMC distance to measure the color difference between pixels. Since the Gabor representation is optimal [21] in the sense of minimizing the uncertainty in the space and the frequency domain, the texture features of pixels are extracted from the image of multiple scales and orientations by using the Gabor wavelet decomposition [22]. When a region A_i is selected or specified as a semantic template, the *visual keyword* ω_i of region A_i can be approximated by the following two steps: (1) spatial feature modeling and (2) color and texture feature modeling.

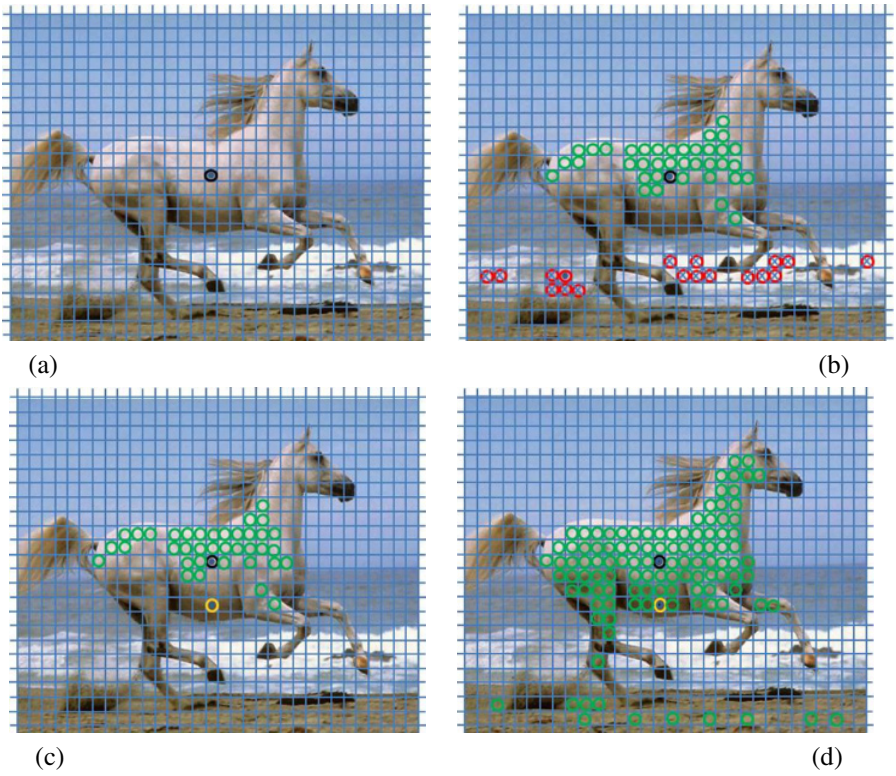


Fig. 1. Example of interactive relevance learning by MGNN. (a): A white horse is used as a designated query region and the marked block contains initial visual features for the MGNN learning; (b): Red and green marked blocks are the verification results when the MGNN learns the selected block in (a), and then another block is selected from the red marked blocks for anti-reinforced learning; (c): Disappear of red check marks in (b) shows the results of anti-reinforced learning, and then a block marked in yellow is selected for reinforced learning of additional visual features in the query region; (d): The reinforcement learning results show that more green marks were added on the abdomen of the white horse.

1) Spatial Feature Modeling:

Given an image region $A_i = \{\mathbf{x}(t) : t = 1, 2, \ldots, N_i\}$, where N_i is the number of pixels in A_i, a 2D Gaussian distribution $p_s(\mathbf{x}(t)|\omega_i)$ can be used to formulate the spatial feature of A_i. Let $p_s(\mathbf{x}(t)|\omega_i, \theta_{s,r_i})$ be a Gaussian cluster to comprise $p_s(\mathbf{x}(t)|\omega_i)$, i.e., $p_s(\mathbf{x}(t)|\omega_i) = \sum_{r_i=1}^{R_i} P_s(\theta_{s,r_i}|\omega_i)\, p_s(\mathbf{x}(t)|\omega_i, \theta_{s,r_i})$, where θ_{s,r_i} represents the parameter set $\{\mu_{s,r_i}, \Sigma_{s,r_i}\}$, and $P_s(\theta_{s,r_i}|\omega_i)$ denotes the prior probability of the

cluster r_i. By definition, $\sum_{r_i=1}^{R_i} P_s(\theta_{s,r_i}|\omega_i)=1$, where R_i is the number of clusters in $p_s(\mathbf{x}(t)|\omega_i)$, and let the cluster r_i be a 2D Gaussian distribution:

$$p_s(\mathbf{x}(t)|\omega_i,\theta_{s,r_i})=\frac{1}{2\pi|\Sigma_{s,r_i}|^{1/2}}\times\exp\left(-\frac{1}{2}\left(\mathbf{x}(t)-\boldsymbol{\mu}_{s,r_i}\right)^T\Sigma_{s,r_i}^{-1}\left(\mathbf{x}(t)-\boldsymbol{\mu}_{s,r_i}\right)\right), \tag{1}$$

where $\boldsymbol{\mu}_{s,r_i}=[\mu_{s,r_i1},\mu_{s,r_i2}]^T$ is the mean vector, and Σ_{s,r_i} is the covariance matrix. *EM* learning algorithm [23] is iteratively applied to adjust the parameters of Eq. (1). There are two steps in each iteration cycle: *Estimation* (E) step and *Maximization* (M) step. The M step maximizes a likelihood function which is further refined in each iteration by the E step. The *EM* learning is to maximize the log likelihood (*E*) of the pixels in A_i, $E=\sum_{t=1}^{N_i}\log(p_s(\mathbf{x}(t)|\omega_i))$. The following update equations adjust the parameters μ, Σ and P_s of the $2D$ mixture Gaussian distribution $p_s(\mathbf{x}(t)|\omega_i,\theta_{s,r_i})$. For the iteration of j:

$$\mu_{s,r_i}^{(j+1)}=\frac{\sum_{t=1}^{N_i}h_{s,r_i}^{(j)}(t)\mathbf{x}(t)}{\sum_{t=1}^{N_i}h_{s,r_i}^{(j)}(t)},\ \Sigma_{s,r_i}^{(j+1)}=\frac{\sum_{t=1}^{N_i}h_{s,r_i}^{(j)}(t)\left(\mathbf{x}(t)-\mu_{s,r_i}^{(j)}\right)\left(\mathbf{x}(t)-\mu_{s,r_i}^{(j)}\right)^T}{\sum_{t=1}^{N_i}h_{s,r_i}^{(j)}(t)},\ P_s^{(j+1)}(\theta_{s,r_i}|\omega_i)=\frac{1}{N_i}\sum_{t=1}^{N_i}h_{s,r_i}^{(j)}(t),$$

$$\text{where } h_{s,r_i}^{(j)}(t)=\frac{P_s^{(j)}(\theta_{s,r_i}|\omega_i)p_s^{(j)}(\mathbf{x}(t)|\omega_i,\theta_{s,r_i})}{\sum_{r=1}^{R}P_s^{(j)}(\theta_{s,r_i}|\omega_i)p_s^{(j)}(\mathbf{x}(t)|\omega_i,\theta_{s,r_i})}.$$

When the *EM* learning converges, and it ideally obtain a maximum likelihood estimation of the image region A_i.

2) Color and Texture Feature Modeling:

As mentioned above, the spatial features of the region A_i in the image I are approximated by the union of R_i elliptic regions. After the spatial modeling, the color and texture feature modeling can be performed in each of the elliptic spatial regions. Suppose that an elliptic region a_{r_i} is approximated by a Gaussian distribution $p_s(\mathbf{x}(t)|\omega_i,\theta_{s,r_i})$. Then, the color features of region a_{r_i} can also be modeled by a Gaussian distribution:

$$p_c(\mathbf{c}_{\mathbf{x}(t)}|\omega_i,\theta_{c,r_i})=\frac{1}{(2\pi)^{D_c/2}|\Sigma_{c,r_i}|^{1/2}}\times\exp\left(-\frac{1}{2}(\mathbf{c}_{\mathbf{x}(t)}-\boldsymbol{\mu}_{c,r_i})^T\Sigma_{c,r_i}^{-1}(\mathbf{c}_{\mathbf{x}(t)}-\boldsymbol{\mu}_{c,r_i})\right),$$

where $\mathbf{c}_{\mathbf{x}(t)}$ is a D_c-dimensional color feature vector at a pixel $\mathbf{x}(t)$, and the mean vector $\boldsymbol{\mu}_{c,r_i}$ and the covariance matrix Σ_{c,r_i} are calculated as follows:

$$\boldsymbol{\mu}_{c,r_i} = \frac{1}{N_{c,r_i}} \sum_{\mathbf{x}(t)\in a_{r_i}} p_s(\mathbf{x}(t)|\omega_i, \theta_{s,r_i}) \mathbf{c}_{\mathbf{x}(t)},$$

$$\Sigma_{c,r_i} = \frac{1}{N_{c,r_i}} \sum_{\mathbf{x}(t)\in a_{r_i}} p_s(\mathbf{x}(t)|\omega_i, \theta_{s,r_i})(\mathbf{c}_{\mathbf{x}(t)} - \boldsymbol{\mu}_{c,r_i})(\mathbf{c}_{\mathbf{x}(t)} - \boldsymbol{\mu}_{c,r_i})^T,$$

where $N_{c,r_i} = \sum_{\mathbf{x}(t)\in a_{r_i}} p_s(\mathbf{x}(t)|\omega_i, \theta_{s,r_i})$.

Since the texture features of the elliptic spatial regions are modeled in the same way as color features, one can obtain the notations and formulas for texture modeling by just replacing the term ***color*** by ***texture*** and subscript c by t in the above description. After the spatial, color, and texture modeling, the joint Gaussian density function is obtained as $p(\mathbf{z}(t)|\omega_i, \theta_{r_i}) = p_s(\mathbf{x}(t)|\omega_i, \theta_{s,r_i}) p_c(\mathbf{c}_{\mathbf{x}(t)}|\omega_i, \theta_{c,r_i}) p_t(\mathbf{t}_{\mathbf{x}(t)}|\omega_i, \theta_{t,r_i})$, where $\mathbf{z}(t) = [\mathbf{x}(t)^T, \mathbf{c}_{\mathbf{x}(t)}^T, \mathbf{t}_{\mathbf{x}(t)}^T]^T$, and $\theta_{r_i} = [\theta_{s,r_i}, \theta_{c,r_i}, \theta_{t,r_i}]$.

Finally, the *visual keyword* ω_i formulates the spatial, color, and texture features of the region A_i by a mixture of Gaussian distributions:

$$p(\mathbf{z}(t)|\omega_i) = \sum_{r_i=1}^{R_i} P(\theta_{r_i}|\omega_i) p(\mathbf{z}(t)|\theta_{r_i}, \omega_i).$$

Therefore, the difference between two visual features can be measured by the difference between two mixture Gaussian distributions.

Due to the distributional forms of the *visual keywords*, the mixture Gaussian based Neural Networks is adopted to model the *visual keywords*. The *MGNN* is a generalized model from its predecessor, *SPDNN* [24]. The *MGNN* contrasts with the *SPDNN* in respect to input data type. The input data of the *MGNN* are in distributional forms instead of numerical forms.

3) Mixture Gaussian Neural Networks Learning:

The schematic of a MGNN is depicted in Figure 2. As similar to its predecessor SPDNN, MGNN has a modular network structure, where one subnet is designated to model one visual feature class.

Assume that the modeled distribution for each class ω_i is $\mathcal{P}_i$, where $i \in \{1, 2, \cdots, k\}$. For an input datum $\mathbf{x}(t)$ with a distribution $\mathcal{P}_t$, the discriminate function of *MGNN* is defined as:

$$\varphi(\mathcal{P}_i, \mathcal{P}_t) = \mathcal{F}(\mathcal{P}_i, \mathcal{P}_i) - 2\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t) + \mathcal{F}(\mathcal{P}_t, \mathcal{P}_t), \text{ where } \mathcal{F}(\mathcal{P}_i, \mathcal{P}_t) = \sum_{n=1}^{R_i} \sum_{m=1}^{R_t} P_n^i P_m^t \mathcal{G}(\theta_n^i, \theta_m^t),$$

$$\text{and, } \mathcal{G}(\theta_n^i, \theta_m^t) = \frac{\exp\left\{-\frac{1}{2}\sum_{d=1}^{D} \frac{(\mu_{m(d)}^t - \mu_{n(d)}^i)^2}{(\sigma_{m(d)}^t)^2 + (\sigma_{n(d)}^i)^2}\right\}}{\sqrt{\prod_{d=1}^{D} 2\pi((\sigma_{m(d)}^t)^2 + (\sigma_{n(d)}^i)^2)}}. \qquad (2)$$

R^D is a D-dimensional feature space; $\mathcal{P}_i = \sum_{n=1}^{R_i} P_n^i p(\mathbf{z}|\theta_n^i)$ and $\mathcal{P}_t = \sum_{m=1}^{R_t} P_m^t p(\mathbf{z}|\theta_m^t)$ are two mixture Gaussian distributions. Here, let * denote i or t, and R_* is the number of mixture components in $\mathcal{P}_*$, P_n^* is the prior probability of the n-th component, $\theta_n^* = \{\mu_n^*, \Sigma_n^*\}$, and $p(\mathbf{z}|\theta_n^*)$ is a multivariate Gaussian with mean vector $\mu_n^* = [\mu_{n(1)}^*, \cdots, \mu_{n(D)}^*]^T$, and covariance matrix $\Sigma_n^* = diag[(\sigma_{n(1)}^*)^2, \cdots, (\sigma_{n(D)}^*)^2]$.

Therefore, the discriminate function is implemented by a two-layer pyramid network. The bottom layer contains three structurally identical pyramid sub-networks, each of which computes the $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_i)$, $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$, and $\mathcal{F}(\mathcal{P}_t, \mathcal{P}_t)$ respectively. Figure 3 depicts the internal architecture of the pyramid sub-network corresponding to $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$. Suppose that the mixture Gaussian distributions $\mathcal{P}_i$ and $\mathcal{P}_t$ consist of R_i and R_t components, respectively. The sub-network for $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$ contains R_i hidden nodes and an $R_i \times R_t$ input nodes, each of which is marked as $G_{n,m}$.

The MGNN adopts the SPDNN learning scheme. While the input datum $\mathbf{x}(t)$ belonging to the class ω_i is misclassified to class ω_j, the reinforced and anti-reinforced learning rules are applied to the subnets of ω_i and ω_j, respectively.

$$\textit{Reinforced Learning rule}: \ \mathbf{w}_i^{(m+1)} = \mathbf{w}_i^{(m)} + \eta \nabla \varphi(\mathbf{x}(t), \mathbf{w}_i) \tag{3}$$

$$\textit{Anti-reinforced Learning rule}: \ \mathbf{w}_i^{(m+1)} = \mathbf{w}_i^{(m)} + \eta \nabla \varphi(\mathbf{x}(t), \mathbf{w}_i) \tag{4}$$

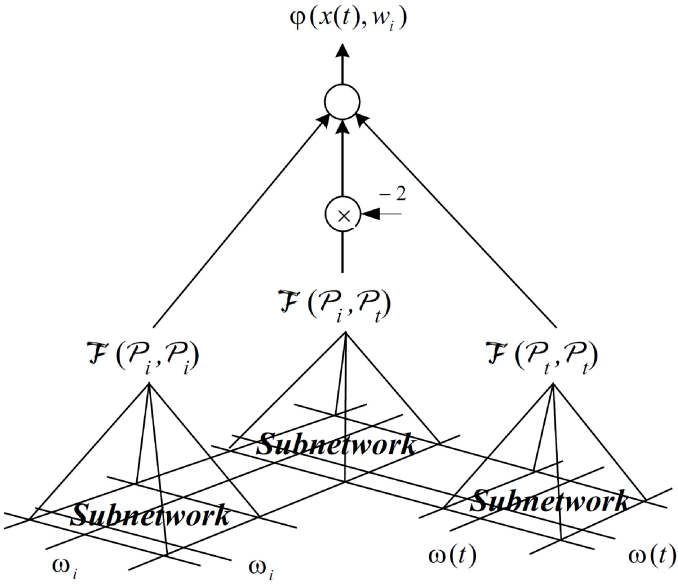


Fig. 2. The schematic diagram of *MGNN*. The detail of a subnet is shown in Figure 3.

The gradient vectors in (3) and (4) are computed as follows: for each $n \in \{1,2,\ldots,R_i\}$ and $d \in \{1,2,\ldots,D\}$:

$$\frac{\partial \varphi(\mathbf{x}(t),\mathbf{w}_i)}{\partial \mu^i_{n(d)}} = 2P^i_n \left[\sum_{m=1}^{R_i} P^i_m \mathcal{N}^i_{m,n,d} (\mu^i_{n(d)} - \mu^i_{m(d)}) - \sum_{m=1}^{R_t} P^t_m \mathcal{N}^t_{m,n,d} (\mu^t_{m(d)} - \mu^i_{n(d)}) \right], \text{ and}$$

$$\frac{\partial \varphi(\mathbf{x}(t),\mathbf{w}_i)}{\partial (\sigma^i_{n(d)})^2} = P^i_n \left[\sum_{m=1}^{R_i} P^i_m \mathcal{N}^i_{m,n,d} \left(\frac{(\mu^i_{m(d)} - \mu^i_{n(d)})^2}{(\sigma^i_{m(d)})^2 + (\sigma^i_{n(d)})^2} - 1 \right) - \sum_{m=1}^{R_t} P^t_m \mathcal{N}^t_{m,n,d} \left(\frac{(\mu^t_{m(d)} - \mu^i_{n(d)})^2}{(\sigma^t_{m(d)})^2 + (\sigma^i_{n(d)})^2} - 1 \right) \right],$$

$$\text{where } \mathcal{N}^x_{m,n,d} = \frac{\mathcal{G}(\theta^x_m, \theta^i_n)}{(\sigma^x_{m(d)})^2 + (\sigma^i_{n(d)})^2}.$$

Procedures and examples of using *MGNN* to learn visual features and retrieve relevance images will be presented and discussed in next section.

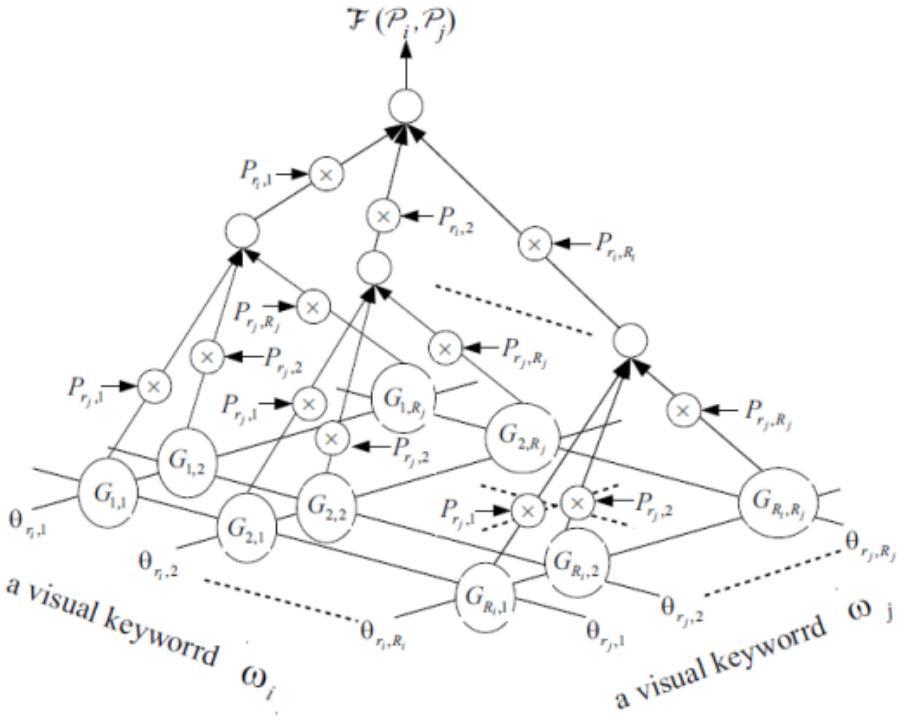


Fig. 3. The internal architecture of a model in Figure 2 for computing of $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_i)$

3 Visual Relevance Image Query

The objective of VRIQ method is to have query and retrieval results fitted to users' desired visual perception. One noticeable feature of the proposed method is to allow users to specify query regions without consciously segmenting images into key objects or regions. More specifically, the proposed method let the user just simply select a small block region, instead of the whole designated regions. Then, the visual features, such as color and texture characteristics of the small block are collected to interactively train a MGNN. When a MGNN completely learns the color and texture features of designated regions, the MGNN would be used to measure the similarity values between the designated regions and images from image database or collections. A flowchart of the proposed VRIQ approach is shown in Figure 4.

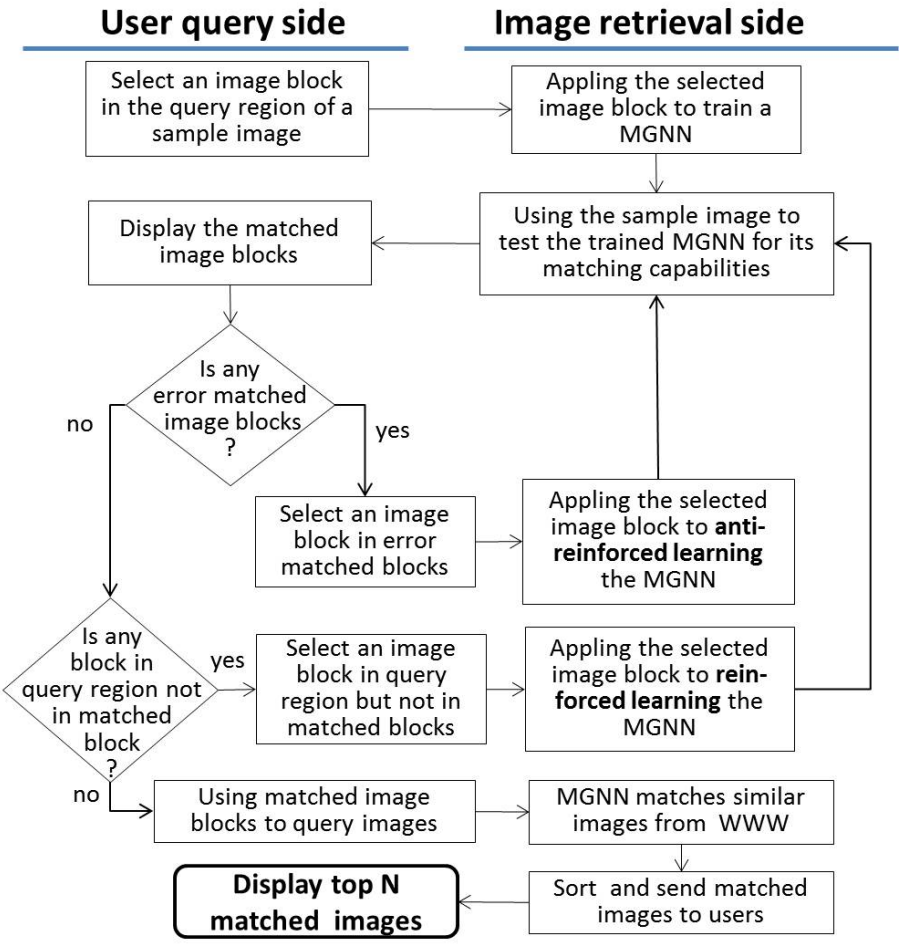


Fig. 4. The flowchart of the proposed MGNN relevance learning and **VRIQ** approach

Figure 5 shows the proposed VRIQ image retrieval results from WWW image collections by using a tiger head image as a designated query region. The image at the top-left corner is the query image, and the rest images are the top query results. As we can see, most of these retrieved results match with the query image, the tiger head.

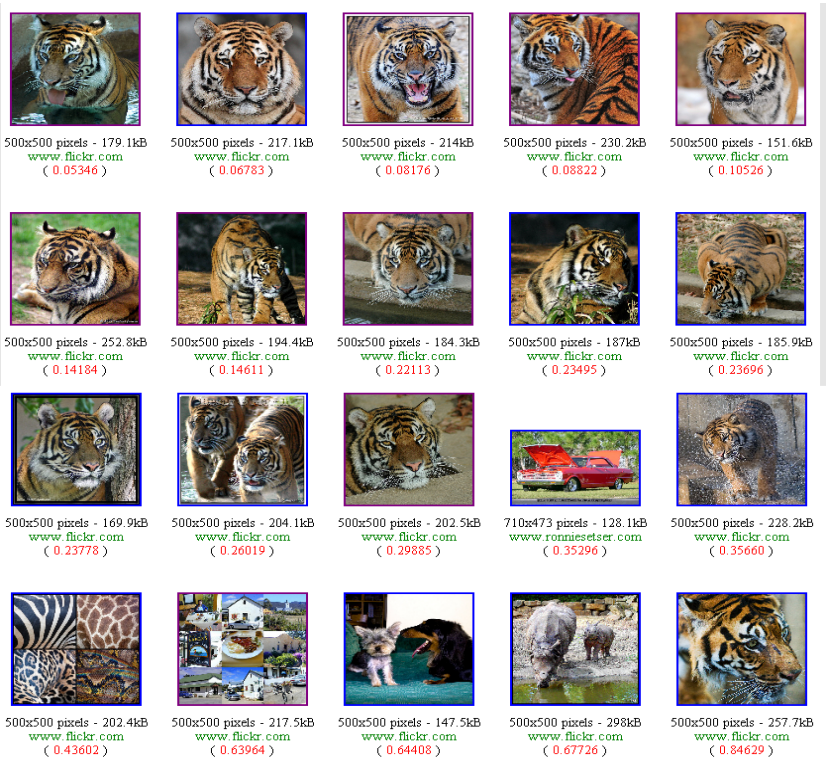
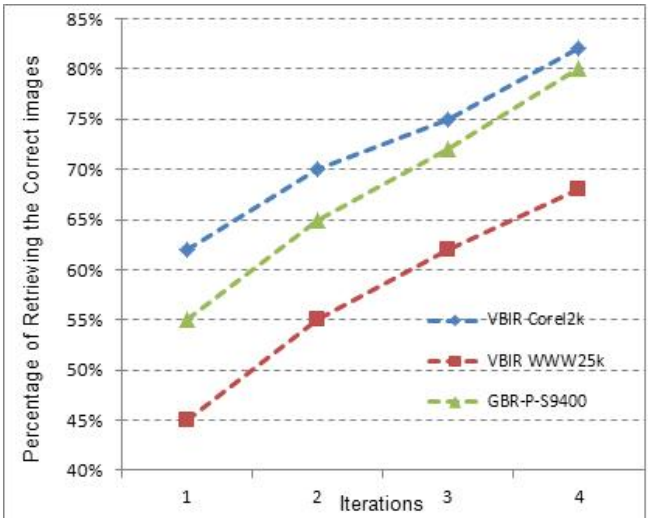


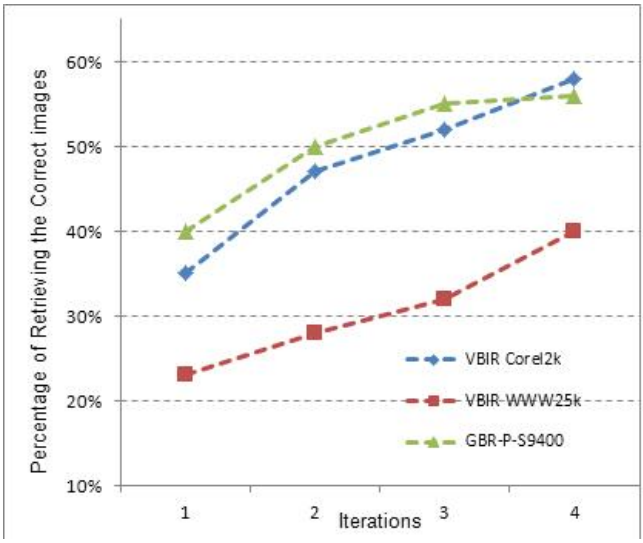
Fig. 5. The retrieval results of the proposed VRIQ approach by query a *tiger head* image as *visual keywords*

4 Experiments and Discussions

Two types of experiments were conducted to evaluate the performance of VRIQ approach. The experiments were performed on two image sets: (1) a set of 2000 images collected from 10 categories of Corel photo gallery, and (2) a set of 25,000 images gathered from WWW. These two image sets are called Corel2k and WWW25k, respectively. The images in Corel2k are labeled in 10 categories, e.g., Africa, beach, building, bus, butterfly, dinosaur, elephant, flower, food, and mountain. Each image in WWW25k was initial labeled by the keywords extracted from surrounding text of the image according to the method proposed by Srihari et al [12]. These categories and labels are considered as the ground truth for verifying the retrieval results.



(a)



(b)

Fig. 6. Performance comparison of image query and retrieval of (a) butterfly, and (b) mountain relevant images among VRIQ Corel2k, VRIQ WWW25k and GBR-P-S9400

The performance was evaluated according to the averaged image retrieval accuracy versus a sequence of query learning. The accuracy [25] is defined as the ratio of relevant images in the top 50 retrieved images. The average accuracy is simply the average of the accuracies measured for the 1600 randomly selected test queries.

Experiment 1- *query on a particular object or category*: Six categories of images *building*, *bus*, *butterfly*, *dinosaur, elephant*, and *flower* were used from both Corel2k and WWW25k image collections. The retrieval performance of query on *butterfly* is shown in Figure 6(a). By repeating four cycles of user relevant feedback, Hsu [26] achieves an 80% of accuracy on a similar type of query on the butterfly category, with over 9400 images selected from a Corel photo gallery. The VRIQ achieves 85% and 68% of accuracies on the butterfly category, with over 2000 images from Corel2k and over 25000 images from WWW25k image collection, respectively. A more interesting comparison would be made if the VRIQ retrieval with comparable image collection were available.

Experiment 2- *query on background scene*: It is very interesting and difficult for an image query system to have capability to retrieve relevant images according to the query on background scenes. The experiments were conducted to retrieve four different types of images, such as *Africa*, *beach*, *food* and *mountain* from both Corel2k and WWW25k image collections. The query results from *mountain* images are shown in Figure 6(b). Similar to Experiment 1, VRIQ approach is slightly superior to the GBR-P-S approach on the Corel image collection, and is inferior to the GBR-P-S approach on the WWW image collection. Notice that the retrieval accuracy of these experiments do not imply that the performance of VRIQ approach is superior or inferior to the GBR-P-S approach [14], since the VRIQ approach uses visual features directly from users' mind, which usually reflect user's designated image regions or objects.

The results show that the proposed VRIQ approach achieved comparable accuracy to the current leading approaches [14]. To summarize the results of the experiments, it appears that (1) the VRIQ approach can retrieve relevant images closely associated with users' query regions or objects, (2) the VRIQ approach has the capability of searching and retrieval relevant images from a large collection such as the WWW25k.

5 Conclusions

In this paper, we introduce an interactive relevance feedback feature learning approach for visual-based image query and retrieval. A comprehensive performance evaluation of the proposed VRIQ approach was made on images from COREL gallery and images gathered from WWW, and comparison with leading approaches [14] was conducted and discussed. From the experimental results, we noticed that the proposed approach (1) can query and retrieve relevance images as close as to users' mind and/or desire, and the closeness is somewhat related to the number of feature leaning iterations, (2) has the capability of searching and retrieving relevant images from large collections. Since mixture Gaussian distributions can be used to approximate various dimensional data, thus the study of using mixture Gaussian neural networks to query and to retrieve high dimensional data or images will be a new area of research for future works.

Acknowledgment. The authors acknowledge Prof. S.Y. Kung, Dr. S.H. Lin for their helpful suggestions regarding the probabilistic DBNN, and statistical pattern recognition methods.

References

1. Shen, H.T., Ooi, B.C., Tan, K.-L.: Giving meanings to www images. In: MULTIMEDIA 2000: Proceedings of the Eighth ACM International Conference on Multimedia, New York, NY, USA, pp. 39–47 (2000)
2. Eakins, J., Graham, M.: Content-based image retrieval. In: Technical Report. University of Northumbria at Newcastle (1999)
3. Sethi, I.K., Coman, I.L.: Mining association rules between low-level image features and high-level concepts. In: Proceedings of the SPIE Data Mining and Knowledge Discovery, vol. III, pp. 279–290 (2001)
4. Chang, S.K., Liu, S.H.: Picture indexing and abstraction techniques for pictorial databases. IEEE Trans. Pattern Anal. Mach. Intell. **6**(4), 475–483 (1984)
5. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., Equitz, W.: Efficient and effective querying by image content. J. Intell. Inf. Syst. **3**(3–4), 231–262 (1994)
6. Pentland, A., Picard, R.W., Scaroff, S.: Photobook: content-based manipulation for image databases. Int. J. Comput. Vision **18**(3), 233–254 (1996)
7. Gupta, A., Jain, R.: Visual information retrieval. Commun. ACM **40**(5), 70–79 (1997)
8. Smith, J.R., Chang, S.F.: VisualSeek: a fully automatic content based query system. In: Proceedings of the Fourth ACM International Conference on Multimedia, pp. 87–98 (1996)
9. Ma, W.Y., Manjunath, B.: Netra: a toolbox for navigating large image databases. In: Proceedings of the IEEE International Conference on Image Processing, pp. 568–571 (1997)
10. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLIcity: semantics-sensitive integrated matching for picture libraries. IEEE Trans. Pattern Anal. Mach. Intell. **23**(9), 947–963 (2001)
11. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys **40**(2), 5:1–5:60 (2008)
12. Srihari, R.K., Zhang, Z., Rao, A., Baird, H., Chen, F.: Intelligent indexing and semantic retrieval of multimodal documents. Information Retrieval **2**, 245–275 (1999)
13. Jing, F., Li, M., Zhang, L., Zhang, H.-J., Zhang, B.: Learning in region based image retrieval. In: Bakker, E.M., Lew, M., Huang, T.S., Sebe, N., Zhou, X.S. (eds.) CIVR 2003. LNCS, vol. 2728. Springer, Heidelberg (2003)
14. Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., Malik, J.: Blobworld: a system for region-based image indexing and retrieval. In: Huijsmans, D.P., Smeulders, A.W. (eds.) VISUAL 1999. LNCS, vol. 1614, pp. 509–517. Springer, Heidelberg (1999)
15. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Trans. Circ. Syst. Video Technol. **8**(5), 644–655 (1998)
16. Xu, Y.-Y., Tseng, C.-L., Fu, H.-C.: Texture Recognition by Generalized Probabilistic Decision-Based Neural Networks. Expert Systems with Applications: An International Journal (0957-4174) **38**(5) (May 2011)
17. Fu, H.-C., Xu, Y.-Y., Pao, H.-T.: Multimodal search for effective image retrieval. In: Proceedings of 15th International Conference on Systems, Signals and Image Processing (IWSSIP 2008), Bratislava, Slovakia, June 25-28, pp. 23–236 (2008)
18. Xu, Y.Y., Chuang, S.C., Tseng, C.L., Fu, H.-C.: Generalized probabilistic decision-based neural networks for texture classification and retrieval. In: The 2010 International Conference on Modelling, Identification and Control (ICMIC), Okayama, Japan, pp. 500–504 (2010)

19. Xu, Y.-Y., Fu, H.-C.: Visual Keyword Based Image Retrieval. in IJCSI 9(3), No 2 (May 2012)
20. Xu, H., Yaguchi, H.: Visual evaluation at scale of threshold to suprathreshold color difference. Color Research and: Application **30**(3), 198–208 (2005)
21. Daugman, J.G.: Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. IEEE Transactions on Acoustics, Speech, and Signal Processing **36**(7), 1169–1179 (1988)
22. Manjuath, B.S., Ma, W.Y.: Texture features for browsing and retrieval of image data. IEEE Transactions on PAMI **18**(8), 837–842 (1996)
23. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B **39**(1), 1–38 (1977)
24. Fu, H.C., Xu, Y.Y.: Multi-linguistic handwritten character recognition by Bayesian decision-based neural networks. IEEE Transactions on Signal Processing **46**(10), 2781–2789 (1998)
25. Su, Z., Zhang, H., Li, S.Z., Ma, S.: Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. IEEE Transactions on Image Processing **12**(8), 924–937 (2003)
26. Hsu, C.T., Li, C.Y.: Relevance feedback using generalized Bayesian framework with region-based optimization learning. IEEE Transactions on Image Processing **14**(10), 1617–1631 (2005)

Scene Classification Based on Local Binary Pattern and Improved Bag of Visual Words

Gholam Ali Montazer[1(✉)], Davar Giveki[2], and Mohammad Ali Soltanshahi[3]

[1] Department of Information Technology Engineering, School of Engineering, Tarbiat Modares University, Tehran, Iran
montazer@modares.ac.ir
[2] Department of Information Technology, Iranian Research Institute for Information Science and Technology (Iran Doc), Tehran, Iran
Giveki@students.irandoc.ac.ir
[3] Department of Computer Science, University of Tehran, Tehran, Iran
ali.soltanshahi@gmail.com

Abstract. Today, image classification is considered as one of the most important and challenging tasks in computer vision. This paper presents a new method for image classification using Bag Of Visual Words and Local Binary Patterns (LBP). The bag-of-visual-words (BoVW) model has been proven to be very efficient for image classification and image retrieval. However, most proposals directly use local features extracted from an image while ignoring hidden information that could be extracted from an image. To solve this problem, we propose a novel image classification method using information extracted from different channels of the image and the grayscale version of the image. In this way more discriminant information is extracted from the image and as a result the constructed BoVW model gives highly discriminative features that considerably increases the classification performance. In this work we embed features extracted using LBP into BoVW model to construct our proposed scene classification model. The choice of LBP as image feature descriptor is because of the fact that the content of most of the scene images contains textural information so extracting LBP features is a very wise choice compared to other popular image features like Scale Invariant Feature Transform (SIFT) that fails to capture image information in homogeneous areas or textual images. Experiments on Oliva and Torralba (OT) dataset demonstrate the effectiveness of the proposed method.

Keywords: Image classification · Bag of visual words · LBP · Scene classification · Bag of features

1 Introduction

The community of machine learning (ML) refers to the word "scene" as a semantically meaningful, nameable human-scaled view of a real world environment [1]. The ability of accurately recognizing and classifying an imaged scene is very important for automatic image analysis systems in a wide variety of tasks. Other than the pure classification of the environment contained in a picture, individuating the scene

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 241–251, 2015.
DOI: 10.1007/978-3-319-19258-1_21

category may help also in object recognition, providing a context on the possible semantic labels of the objects identities (e.g., a fish is rare to see in a desert environment) [2]. Conversely, as reported in [3], a human being does not need to perceive the objects in a scene to identify its semantic category: behavioral and computational studies show that humans rely on global visual properties to exploit scene classification, instead of performing recognition of particular objects in a scene. Recent published works show that a usual approach for scene classification is to extract global features to represent the scene. The basic idea behind these approaches is to take the whole image as an entity then relies on low-level features (e.g. color [4,5], edge response [5], texture [4], gradient, etc.) to represent the characteristics of the scene. Authors in [6] proposed a global feature called 'gist' that employs a visual attention model to combine global color, intensity and orientation features to represent the scene. This may be sufficient for separating scenes with significant differences in the global properties. However, if scenes with similar global characteristics (e.g. bedroom vs. sitting room) are to be differentiated, then global features may not be discriminative enough. Thus, features extracted from local regions in a scene have been proposed for classification [7,8]. Therefore, representing an image by a collection of local image patches of certain size using unsupervised learning methods [9–11] has become very popular and achieved great success in visual recognition, image retrieval, scene modeling/classification, etc., because of its robustness to partly occlusions, scaling, translation and illumination variations. In scene classification, the scene type is represented by the co-occurrences of a large number of visual components or the co-occurrences of a certain number of visual topics [12,13]. In these proposed methods, local image features are quantized into a set of visual words to form a codebook. Then an image is represented by the distribution of the visual words in the codebook with or without geometric configurations. Fei-Fei and Perona [12] and Quelhas et al. [13] independently proposed two different unsupervised learning methods to learn visual words from local regions of the scene images, from which the distributions of the visual words are used to represent the images. Furthermore, a latent variable called 'theme' [12] is also learned and taken as the intermediate representation of the scene. Lazebnik et al. [14] proposed a spatial pyramid matching based method for scene classification. This method calculates the distribution of the visual words at multi-spatial resolutions to form a spatial pyramid representation of an image, then employs the pyramid matching strategy proposed by Grauman [15] to measure the similarity between pyramids. A comparative study conducted by Bosch et al. [16] has pointed out that using visual words jointly with different techniques, such as the probabilistic latent semantic analysis (pLSA)[13,17], latent Dirichlet allocation [12], is the one that can achieve the best classification results for scene classification. By reviewing the literature, it is noted that the visual words proposed previously, however, only takes into account local image features extracted from the original image and not from the different possible types of the image. For instance, one way is that, an RGB image can be first decomposed into three different channels (namely R, G and B), then local image features from every channel can be extracted to form the codebook. In addition, an image can be converted from one color space to another color space in which more powerful and discriminative image features can be extracted. Hence, in this paper we propose a new method for scene classification with capturing more discriminant information to generate histogram of words. To extract image features we densely applied Local Binary Patterns on each color channel.

Choosing LBP as image descriptor highly affects the classification performance because this descriptor is able to effectively capture textural information of an image and we know that most of the scene images in the dataset are composed of certain textures. The organization of the rest of the paper is as follows. Section 2 gives an overview about Bag of Visual Words model. In section 3 the proposed classification method is discussed. Dataset description and experimental results are reported in section 4 and finally conclusion is drawn in section 5.

2 Bag of Visual Words Model

In the recent years one can find in the literature on image classification and image retrieval, an increasing number of work which make use of the bag of visual words method to represents an images using histograms of quantized features of local patches [18-23]. The BoVW model was first proposed for text document analysis and further adapted for computer vision applications [24]. Studies have shown that local features represented by BoVW are suitable for image classification with wonderful performance [12,14]. Constructing the BoVW from the images involves the following steps: (1) Automatically detect regions/points of interest (local patches) and compute local descriptors over these regions/points, (2) quantize the descriptors into words to form the visual vocabulary using a clustering technique like k-means clustering, (3) count the occurrences of each specific word in the vocabulary for every image in the dataset in order to build the histogram of words and (4) Learn the histogram of words using a classifier. Fig. 1 schematically describes the four steps involved in the definition of the BoVW method.

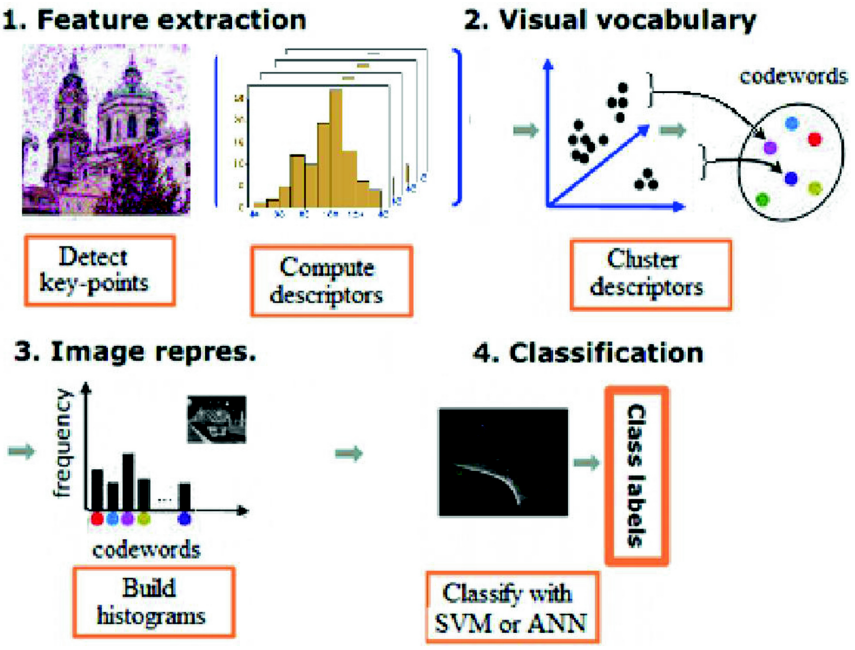


Fig. 1. Four steps to compute the bag-of- visual-words

Feature representation methods, like Scale-invariant feature transform [25], express the image patches as numerical vectors. SIFT represents each patch to 128-dimensional vector. Then each image is considered as a collection of vectors of the same dimension (128 for SIFT) without considering the order of different vectors.

The simplicity of BoVW and requiring relatively small amount of supervision are two important factors that increase the popularity of this method. For labeling training data we just need to determine the class of the image. After determining the class of the image one can easily do further processes, for example in the case of image retrieval, one is interested in retrieving similar images from the same category.

3 Proposed Method

LBP has been proven to be one of the best texture methods available today [26]. It is invariant to monotonic changes in gray-scale and fast to calculate. Its efficiency originates from the detection of different micro patterns (edges, points, constant areas etc.) [26]. Previous research which used uniform patterns representing the most essential texture information showed a strong discriminative ability [26].

LBP represents the description of pixel vicinity image in the binary form. Basic LBP operator uses eight pixels of vicinity, accepting the central pixel as a threshold (see Fig. 2). Pixels with the values, higher than the central one (or equal to it), accept the value <1>, those which are lower than the central one, accept the value <0>. Thus, we get the eight-bit binary code, which describes the pixel vicinity [27].

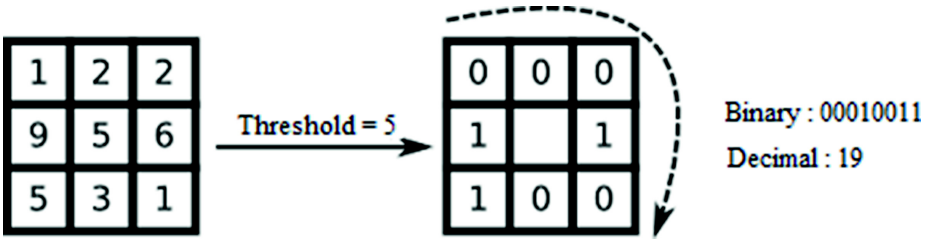


Fig. 2. Basic LBP operator

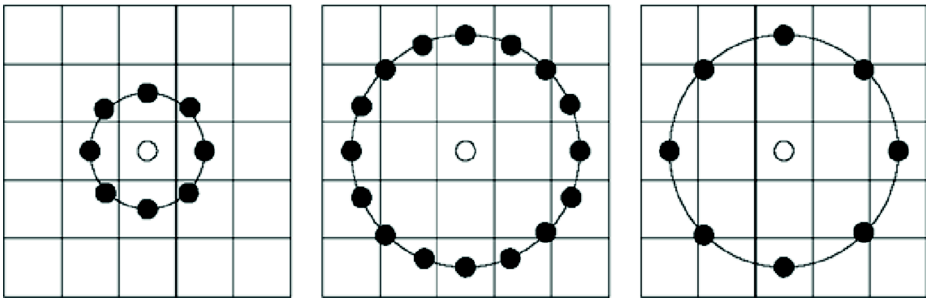


Fig. 3. Neighborhood set for different (P, R). The pixel values are bilinearly interpolated

Certain binary codes contain more information than others. A local binary pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa. For example, 00000000, 001111000 and 11100011 are uniform patterns. Thus, it is possible to use only a subset of all Local Binary Patterns (P(P-1)+2 patterns instead of 2^P) to describe the texture of images. Uniform LBP determine only important local textures, such as ends of lines, edges, angles, spots. Also it provides the economy of memory. In case of $LBP_{8,1}^{u}$, (8 and 1 are respectively the number of neighboring sample points and the radius of $LBP_{P,R}$ and u means uniform pattern), we get a feature vector of 58 bins. As the remaining, un-uniform patterns are accumulated to a single bin, the uniform LBP histogram actually contains 59 bins. To extract LBP features we put a window size of 16×16 pixels around each pixel and compute the uniform pattern $LBP_{8,1}^{u_j}$ of each pixel. These descriptors are composed as a 64-dimensional vector: $LBP_i = [LBP_{8,1}^{u_1}, LBP_{8,1}^{u_2}, \dots, LBP_{8,1}^{u_{64}}]$. In order to extract LBP features, at first, Grey level images are normalized to have intensities with mean zero and unit standard deviation. Color images are first normalized as in "Gray World" to have R, G and B components $R * (\mu/\mu_r), G * (\mu/\mu_g), B * (\mu/\mu_b)$ where $\mu = (\mu_r + \mu_g + \mu_b)/3$ and μ_r, μ_g and μ_b are the mean of each component. The HSV is then computed from these normalized values. For extracting more discriminant features as it was stated before, each image is converted from RGB color space to HSV and HSI color spaces and then in each color space, different color channels, namely R, G, B, H, S, V and I are separated. Next, LBP features are extracted from the whole image at once and from each color channel.

Afterwards, new feature vectors are constructed using concating three various vectors obtained from H, S, V, I, R, G and B channels. Therefore, in total we will have a 177- dimensional feature vector after concatenating the LBP features extracted from each different color channel. Then we perform k-means clustering of random subset descriptors to form a visual vocabulary of visual words. Each descriptor is quantized into a visual word using the nearest cluster center. For considering more spatial information into account, we follow the scheme proposed by Lazebnik et al. [14] which is based on spatial pyramid matching. Consider matching two images each consisting of a 2D point set, where we wish to determine soft matches between the point sets when the images are overlaid for a particular point the strength of the match depends on the distances from its position to points in the other set. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction (like a quadtree).

The number of points in each grid cell is then recorded. This is a pyramid representation because the number of points in a cell at one level is simply the sum over those contained in the four cells it is divided into at the next level. The cell counts at each level of resolution are the bin counts for the histogram representing that level. The soft correspondence between the two point sets can then be computed as a weighted sum over the histogram intersections at each level. Similarly, the lack of correspondence between the point sets can be measured as a weighted sum over histogram differences at each level. In the image case, the pyramid matching is applied to the two-dimensional image space, and a BoVW vector is computed for each grid cell at each pyramid resolution level. So, the 2D points in the example above are replaced by visual words obtaining a Pyramid Histogram Of visual Words descriptor for the

image. In forming the pyramid the grid at level l has 2^l cells along each dimension. Consequently, level 0 is represented by a V-vector corresponding to the V visual words of the histogram, level 1 by a 4V-vector etc, and the Pyramid Histogram Of visual Words descriptor of the entire image is a vector with dimensionality $V\sum_{l\in L} 4^l$.

4 Dataset Description and Experimental Results

4.1 Dataset Description

Oliva and Torralba (OT) dataset: Oliva and Torralba [28] dataset includes 2688 images classified in 8 categories: 360 coasts, 328 forest, 260 highway, 308 inside of cities, 374 mountain, 410 open country, 292 streets and 356 tall buildings. Note that river and forest scenes are all considered as forest, moreover there is not a specific sky scene since almost all of the images contain the sky object. These annotations make a higher inter-class variability. Most of the scenes present a large intra-class variability. The average size of each image is 250×250 pixels.

4.2 Experimental Results

In all of the experiments the Multi-Class Support Vector Machine (M-SVM) classifier is used. For the SVM classifier an exponential kernel of the form $\exp - \alpha d$ is applied, where d is the Euclidean distance between the vectors, and the scalar α is determined as described in [29] (with the trade-off between training error and margin at $C = 1$). The multi-class classification is done using the one-versus-all rule: a classifier is learned to separate each class from the rest, and a test image is assigned the label of the classifier with the highest response. 60% of whole dataset were used for training and the rest were used for testing the performance of the proposed method.

In order to compare the efficiency of our proposed method with that of other methods we used popular statistical measure namely Mean Accuracy, mean Average Precision (mAP) and Receiver Operating Characteristic curve (ROC curve). Table 1 shows the results of implementing the proposed method on OT dataset using different concatenation of feature vectors.

Table 1. Performance of different features on OT dataset

Our Proposed Method	Mean Average Precision	Mean Accuracy	Length of the feature vector
LBP-SVM-RGB	87.84	83.4	512
LBP-SVM-HSV	81.46	78.64	512
LBP-SVM-IGB	87.92	83.2	512
LBP-SVM-IHV	88.53	84	512
LBP-SVM-HVB	**88.88**	**84.8**	512

From Table 1, it can be seen that using 512 visual words and concatenating H, V and B color channels our proposed method achieves the highest performance. It should be noted that, although we tested all the possible combinations, in Table 1 we just reported some of them.

In addition, in order to give more insight into the behavior of the proposed classification method, confusion matrix of the proposed method is reported.

The confusion matrix shows that the proposed image classification method is confused between mountain and open country, main reason for this is that, these two categories semantically are very close to each other and also huge portion of these images are composed of similar patches like sky.

From Fig. 4 we see that our proposed descriptor works very well for forest with accuracy of 94.7% and tall buildings with 91.6% of accuracy. The reason for this, is that, most of the images in these categories are textural image. On the other hand, this descriptor does not provide very promising results for highway with 79.8% of accuracy and open country with 76.2% of accuracy.

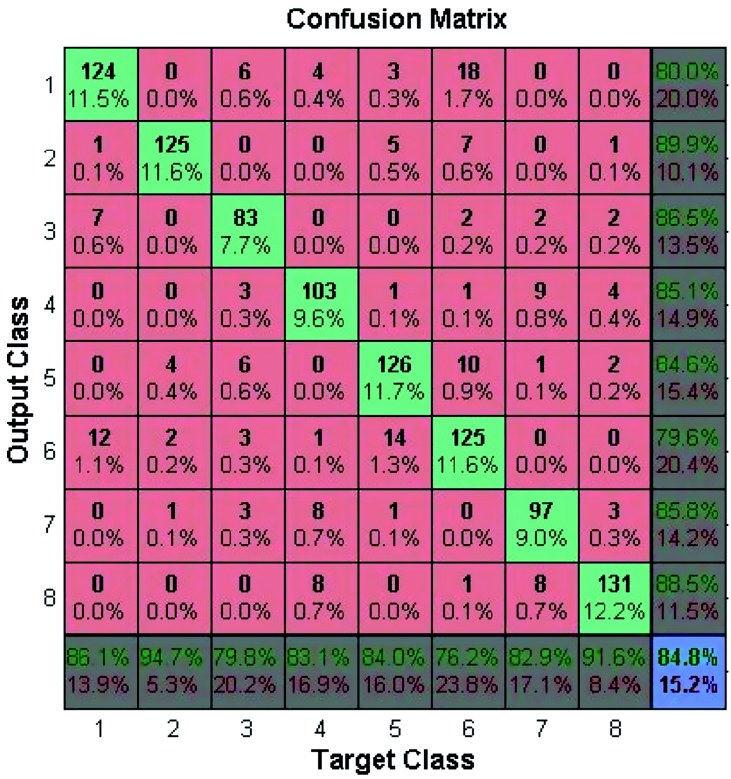


Fig. 4. Confusion matrix of our proposed method using LBP-SVM-HVB

Fig. 5 shows the ROC curve of the proposed method using H,V and B channels for each class in OT dataset. We have also depicted Average Precision for each class in OT dataset in Fig. 6.

Additionally, we compare our proposed method with some state-of-the-art methods. In [30] Wang et al described a new approach for searching images from a given scene image dataset, and introduced a scene image description called packed dense interest points. In the first step of search, semantic retrieval returns a set of candidate images that reflect the intention of the user. The second step re-ranks asymmetric matching based on

the dense packed interest point that serve as constraints on the candidate images. They used 240 images for training (i.e. 30 images per category) and left the rest as test set. Table 2 gives results of implementing other methods on OT dataset.

As it can be seen from the Table 2, compared to other methods our proposed method achieves very good results with 84.8% precision. This fact is because of the high discriminative power of the new image descriptor.

Table 2. Comparison of our proposed method to other methods on OT dataset

Method	mAP
Our Proposed Method Using LBP-SVM-HVB	**84.8%**
Proposed Method in [14]	72%
Proposed Method in [31]	59.2%
Proposed Method in [30]	77.4%
Proposed Method in [32]	63.5%
Proposed Method in [33]	67.4%

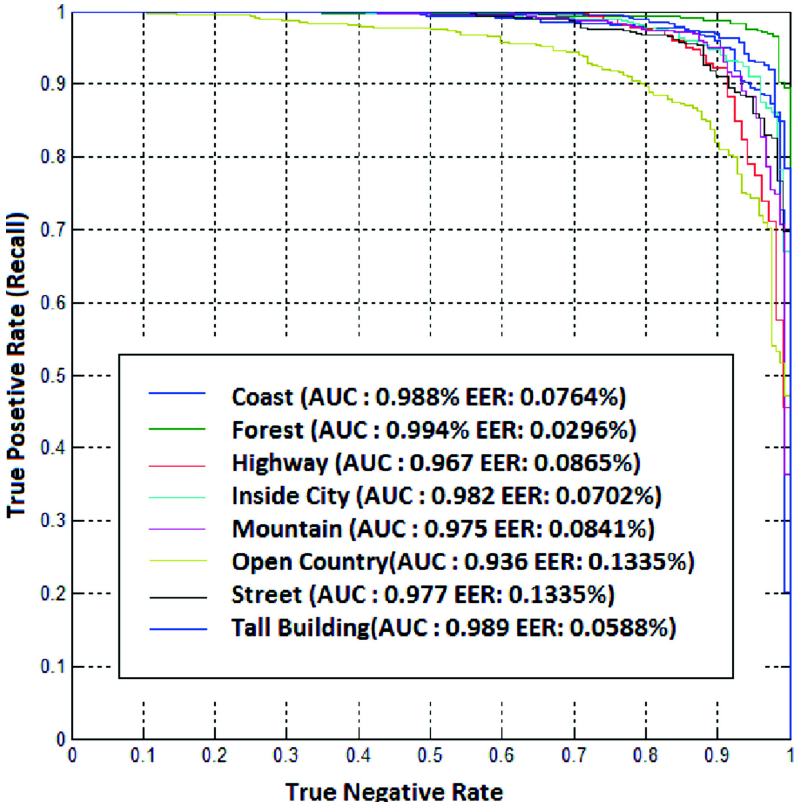


Fig. 5. ROC curves for each class in OT dataset

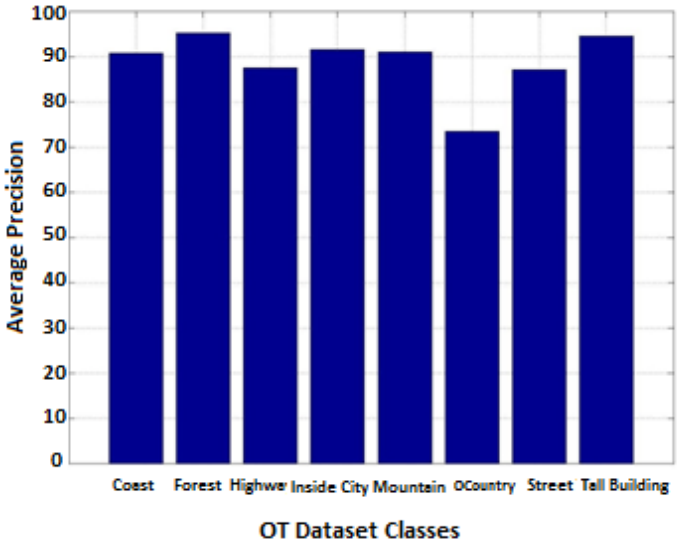


Fig. 6. Average Precision for each class in OT dataset

5 Conclusion

This paper addresses scene classification problem by using features extracted from Local Binary Patterns algorithm and Bag of Visual Words method. The novelty here is that, instead of extracting image features from the whole image, we investigate extracting image features from different image channels in this way the extracted features are more discriminant because more information can be captured from different channels in contrast to just one channel. The experimental results on OT dataset show that our proposed method achieves promising results compared to other state of the arts methods while feature vectors are constructed by concatenating H (Hue), V (Value) B (Blue) channels.

References

1. Henderson, J.: Introduction to real-world scene perception. Visual Cognition **12**(3), 849–851 (2005)
2. Heitz, G., Koller, D.: Learning spatial context: using stuff to find things. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 30–43. Springer, Heidelberg (2008)
3. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. Progress in Brain Research: Visual Perception **155**, 23–36 (2006)
4. Chang, E., Kingshy, G., Sychay, G., Gang, W.: Content-based soft annotation for multimodal image retrieval using Bayes point machines. IEEE Transactions on Circuits and Systems for Video Technology **13**(1), 6–38 (2003)

5. Vailaya, A., Figueiredo, M., Jain, A., Zhang, H.J.: Content-based hierarchical classification of vacation images. In: IEEE International Conference on Multimedia Computing and Systems, vol. 1, pp. 518–523 (1999)
6. Siagian, C., Itti, L.: Gist: a mobile robotics application of context-based vision in outdoor environment. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 3, pp. 1063–1069 (2005)
7. Torralba, A., Oliva, A.: Statistics of natural image categories. Network: Computation in Neural Systems **14**(3), 391–412 (2003)
8. Vogel, J., Schiele, B.: A semantic typicality measure for natural scene categorization. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 195–203. Springer, Heidelberg (2004)
9. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: Tenth IEEE International Conference on Computer Vision, vol. 2, pp. 1816–1823 (2005)
10. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(11), 1475–1490 (2004)
11. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: Ninth IEEE International Conference on Computer Vision, pp. 1470–1477 (2003)
12. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 524–531 (2005)
13. Quelhas, P., Monay, F., Odobez, J.M., Gatica-Perez, D., Tuytelaars, T., Van Gool, L.: Modeling scenes with local descriptors and latent aspects. In: Tenth IEEE International Conference on Computer Vision, vol. 1, pp. 883–890 (2005)
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2169–2178 (2006)
15. Grauman, K., Darrell, T.: The pyramid match kernel: discriminative classification with sets of image features. In: Tenth IEEE International Conference on Computer Vision, vol. 2, pp. 1458–1465 (2005)
16. Bosch, A., Munoz, X., Marti, R.: Which is the best way to organize/classify images by content? Image and Vision Computing **25**(6), 778–791 (2007)
17. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via pLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
18. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning, pp. 1–22. ECCV Computer Vision (2004)
19. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for scene detection. In: European Conference on Computer Vision, vol. 2, pp. 575–588 (2006)
20. Yu, J., Qin, Z., Wan, T., Zhang, X.: Feature integration analysis of bag-of-features model for image retrieval. Neurocomputing **120**, 355–364 (2013)
21. Penatti Otávio, A., Silva, B., Fernanda, B., Valle, E., Gouet-Brunet, V., Torres, R.da S.: Visual word spatial arrangement for image retrieval and classification. Pattern Recognition **47**, 705–720 (2014)
22. Zhang, S., Tian, Q., Hua, G., Huang, Q., Gao, W.: ScenePatchNet: Towards scalable and semantic image annotation and retrieval. Computer Vision and Image Understanding **118**, 16–29 (2014)

23. Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2126–2136 (2006)
24. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision **43**(1), 29–44 (2001)
25. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
26. Ojala, D., Pietikäinen, M., Mäenpää, T.: Multiresolution gray scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 971–987 (2002)
27. Maenpaa, T.: The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications. Oulu University Press (2003)
28. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision **42**(3), 145–175 (2001)
29. Zhang, J., Marszałek, M., Lazebnik, C., Schmid, S.: Local features and kernels for classification of texture and Scene categories: a comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)
30. Wang, H., Liang, W., Wu, X., Teng, P.: Scene image retrieval via re-ranking semantic and packed dense interest points. Neurocomputing **119**, 65–73 (2013)
31. Kim, J., Grauman, K.: Asymmetric region-to-image matching for comparing images with generic object categories. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2344–2351 (2010)
32. Dai, D., Wut, T., Zhu, S.: Discovering scene categories by information projection and cluster sampling. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3455–3462 (2010)
33. Wang, H., Teng, P., Liang, W.: Packed dense interest points for scene image retrieval. In: Sixth IEEE International Conference on Image and Graphics (ICIG), pp. 789–794 (2011)

An Experimental Comparison for the Identification of Weeds in Sunflower Crops via Unmanned Aerial Vehicles and Object-Based Analysis

María Pérez-Ortiz[1(✉)], Pedro Antonio Gutiérrez[2], Jose Manuel Peña[1], Jorge Torres-Sánchez[1], César Hervás-Martínez[2], and Francisca López-Granados[1]

[1] Institute for Sustainable Agriculture, CSIC, P.O. Box 4084, 14080 Córdoba, Spain
i82perom@uco.es
[2] Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein building, 14071 Córdoba, Spain

Abstract. Weed control in precision agriculture refers to the design of site-specific control treatments according to weed coverage and it is very useful to minimise costs and environmental risks. The crucial component is to provide precise and timely weed maps via weed monitoring. This paper compares different approaches for weed mapping using imagery from Unmanned Aerial Vehicles in sunflower crops. We explore different alternatives, such as object-based analysis, which is a strategy that is spreading rapidly in the field of remote sensing. The usefulness of these approaches is tested by considering support vector machines, one of the most popular machine learning classifiers. The results show that the object-based analysis is more promising than the pixel-based one and demonstrate that both the features related to vegetation indexes and those related to the shape of the objects are meaningful for the problem.

Keywords: Unmanned Aerial Vehicles · Object-based analysis · Weed mapping · Image segmentation · Support vector machines

1 Introduction

It is well-known that weeds are responsible for a large reduction in potential global crop yields (approximately a 35%). Because of this, nowadays, most farmers in the EU rely on synthetic herbicides as an useful tool for maintaining and

This work was financed by the Recupera 2020 Project (Spanish MINECO and EU-FEDER Funds). Research of Mr. Torres-Sánchez and Dr. Peña was financed by the FPI and Ramón y Cajal Programs, respectively. Research of Dr. Gutiérrez and Dr. Hervás-Martínez has been subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain).

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 252–262, 2015.
DOI: 10.1007/978-3-319-19258-1_22

ensuring the quality and quantity of crop production, which usually provides a weed control efficacy of 75%.

There are however very clear economical and environmental risks derived from over application, due to herbicides are applied to the whole field even when weeds are distributed in patches. The cost of these herbicides usually accounts for 40% of the cost of all of the chemicals applied to agricultural land in Europe [1] and this economic factor together with environmental concerns have led to the creation of the European legislation on the sustainable use of pesticides, which includes several specific guidelines for reduction of herbicides according to the weed coverage [2,3]. This has been possible because of the introduction of patch spraying in the machinery used, which has enabled the feasibility of site-specific weed management (SSWM) based on weed coverage. A key component of SSWM is to provide precise and timely weed maps for an appropriate early post-emergence weed control, and one of the crucial steps for weed mapping is weed monitoring, either by ground sampling or by remote detection of weeds.

Nonetheless, in early growth stages, the spectral and appearance characteristics of both crops and weeds are similar, thus imposing additional difficulties for the detection. Most previous works have addressed this problem by mapping weeds at late growth stage (e.g. flowering) using piloted aircrafts or QuickBird satellite imagery [4,5]. However, this technology is not suitable in early detection because of the scarce spatial resolution that these kind of platforms provides (pixel size around 50cm and 2.6m for piloted aircrafts and QuickBird satellite, respectively). Recently, a new aerial platform has joined the traditional ones, known as the Unmanned Aerial Vehicle (UAV) [6]. Different studies have highlighted the advantages of UAVs over airborne or satellite equipment [7,8], specially a minor cost, a higher flexibility in flight scheduling and a better spacial resolution. These advantages make UAVs a proper tool to perform multi-temporal studies for crop and weed monitoring at early crop and weed phenological stage [9,10], which is a classic limitation of the traditional remote-sensed platforms.

In many cases, image information is represented by means of low-level features. However, a significant degree of abstraction can be achieved if an appropriate representation of the data is used [11,12]. In this sense, this paper compares different approaches to deal with the problem of weed mapping via remote sensing. These weed maps have to be provided timely, therefore, time is one of the factors to analyse in this study. This paper tries to approach some of the hypotheses and issues that have been identified when using a object-based analysis [11] of the images. Roughly speaking, object-based analysis is devoted to the division of remote sensing imagery into meaningful sets of pixels (known as objects) which are considered as similar based on some measure of homogeneity. The use of object-based analysis methods is spreading rapidly because of the advantages that have been seen when comparing this approximation to the common pixel-based approach. The first hypothesis that we try to validate in this study is whether an object-based analysis is the best option, and how large is the performance gap between such a method and a pixel-based approach. As a second

hypothesis, we validate the use of histograms when using an object-based approach as opposed to the use of different statistical metrics. Finally, we also test different sets of features and training data, in order to analyse the usefulness of these features and the number of patterns needed to yield a reasonable performance. Note that although the base problem of this paper could be addressed from a binary classification point of view, crop detection is also a very important challenge for a wide range of applications, such as plant counting, detecting sowing failures or positioning the patch spraying equipment according to crop rows. Therefore, we consider the discrimination of weeds, crops and bare soil. To conduct this study, different datasets have been created from a sunflower field naturally infested by weeds (considering the previously mentioned hypotheses) and the results in each case are validated by the use of the well-known Support Vector Machine classifier.

The paper is organised as follows: Section 2 shows a description of the data acquisition stage; Section 3 describes the different approaches tested; Section 4 presents the experimental study and analyses the results; and finally, Section 5 outlines some conclusions and future work.

2 Data Acquisition and Processing

The UAV system was tested in a sunflower field situated at the private farm La Monclova, in La Luisiana (Seville, southern Spain, coordinates 37.527N, 5.302W, datum WGS84). The set of aerial images were collected on May 15^{th} 2014, just when post-emergence herbicide or other control techniques are recommended in this crop. The sunflower was at the stage of 4-6 leaves unfolded [13]. The sunflower field was naturally infested by weeds and they had a similar size or, in some cases, were smaller than the crop plants. An experimental plot of 100×100m was delimited within the crop-field to perform the flights. The coordinates of each corner of the flight area were collected using a global position system (GPS) to prepare the flight route in the mission-planning task.

2.1 UAV and Sensor Specifications

A quadrocopter platform with vertical take-off and landing, model md4-1000 (microdrones GmbH, Siegen, Germany), was used to collect the set of aerial images over the above-mentioned experimental crop-field. The flight route was programmed into the UAV software to allow the UAV to reach the programmed altitude and required degree of image overlapping for further mosaicking. The imagery was collected at the altitude of 30 meters. A low cost still camera, model Olympus PEN E-PM1 (Olympus Corporation, Tokyo, Japan) was used. At the moment of each shoot, the on-board computer system records a timestamp, the GPS location, the flight altitude, and vehicle principal axes (pitch, roll and heading). The Olympus camera acquires 12-megapixel images in true colour (R, G and B bands) with 8-bit radiometric resolution and is equipped with a 14-42 mm zoom lens. A sequence of 60% end or longitudinal lap and 30% side or lateral lap imagery was collected to cover the whole experimental sunflower field.

2.2 Image Mosaicking

As said, different overlapped images were collected for this study to cover the whole experimental field. This is due to UAV images flying at low altitudes that can not cover the whole field, and this causes the need to take a sequence of multiple overlapped (end-lap or lateral-lap and side-lap or longitudinal-lap) images. As consequence, a large number of UAV images were acquired to cover the whole sunflower plot. A necessary step is the combination of these individual images via a process of image orthorectification and mosaicking. The Agisoft Photoscan Professional Edition (Agisoft LLC, St. Petersburg, Russia) software was employed for this task.

2.3 Object-Based Image Analysis

Object-based image analysis (OBIA) can be said to be a sub-discipline of the research field of geographic information systems. OBIA is mainly devoted to divide remote sensing imagery into meaningful objects by assessing their characteristics. Objects are image regions derived by one or more criteria of homogeneity in one or more dimensions (i.e., characteristics of the feature space). Therefore, several advantages of objects over single pixels have been found [11]: objects can entail further information apart from the spectral characteristics (i.e., deviation of the values per band, shape, texture, relations with other objects, etc), the use of objects as basic units reduces the computational load of the classification method and also enables the user to consider more complex techniques (as the one proposed in this paper), and finally, that image objects could help to overcome the so called "salt and pepper effect".

The segmentation algorithm used in this case is the one implemented in [14] given that it is mainly focused on the remote sensing setting and designed for large-scale data. This method is based on the generation of spectrally similar units with a minimum object size. The procedure uses the well-known k-means clustering methodology [15] and a object-refining step (to avoid very small objects). The algorithm presents two key parameters: the number of clusters k, and the minimum object size. These parameters have been optimised using the Johnson and Xie method [16] for measuring segmentation quality.

The result of this procedure can be seen in Fig. 1 for a region of the experimental field. Note that each object has been represented in a colour (by averaging the spectral values of all the pixels of the object).

2.4 Data Labelling

To extract the data from the three different classes (crop, soil and weed), we have randomly selected and labelled different objects until the total number per class was n (in this case, we chose $n = 100$). This procedure is supervised by an expert in the topic. As said, the choice of these patterns was completely random. Therefore, depending on the results, these could indicate the necessity of developing more intelligent techniques for choosing the training patterns or validate the fact that a random selection represents a suitable choice in this case.

Fig. 1. The image on the left shows a region selected from the experimental field. The image on the right shows the output of the segmentation algorithm (note that objects have been plotted as regions with the same spectral information by measuring their spectral values).

3 Pattern Classification

The goal in classification is to assign an input vector $\mathbf{x}$ to one of k classes (this label will be designed as y, where $y \in \mathcal{Y} = \{1, \ldots, k\}$), when considering an input space $\mathcal{X} \in \mathbb{R}^d$, where d is the data dimensionality. The training data are assumed to be generated from an i.i.d. $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ from an unknown distribution $P(\mathbf{x}, y)$. Therefore, the objective in this type of problem is to find a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y},\ f \in \mathcal{F}$ that minimises the expected loss or risk.

In this paper, we compare two different approximation for the problem of weed mapping by means of machine learning classifiers: pixel-based and object-based analysis.

3.1 Pixel-Based Approach

Concerning the pixel-based approximation, we propose two different approaches:

- Firstly, to extract the corresponding set of pixels from each labelled object. Each of these pixels will be considered as a new pattern. The label of the pixel will be the one associated to each object.
- Secondly, to reduce each object to a single pixel. To do so, we randomly select one of the pixels of the object.

The previous approaches would try to validate whether a pixel-based approach is suitable, and whether each object could be simplified by randomly choosing one of its components. In this case, only the spectral characteristics are used as input features for the model.

3.2 Object-Based Approaches

In relation to the object-based techniques, we present two different techniques depending on how the patterns are composed:

- Firstly, we would like to test the most common approximation for this type of problem, i.e., the simplification of each object by considering some specific statistical measures for the corresponding pixels. In this case, we consider the mean and standard deviation for each channel.
- Secondly, we propose to use histograms formed by the pixels of the object for each different channel. In this case, we simplify each histogram by grouping the elements into 25 different equally-spaced bins.

Features Considered. As stated before, OBIA methods present the advantage that they can entail further information apart from the spectral one. In this subsection, we will present the different features considered in this paper:

1. Spectral information: In this case we include the statistical metrics or the complete set of values via a histogram representation for each of the three channels (i.e., red, green and blue).
2. Information related to vegetation indexes: In our case, we consider the well-known Excess Green index [17]. We include two different sets of features in this case: vegetation indexes associated to the object itself, and vegetation indexes for the adjacent objects. For the latter case, we analyse the neighbouring objects to the one considered. Fig. 2 shows how we computed neighbouring objects, i.e. by selecting adjacent pixels and storing their object identifier. Then, we compute the mean vegetation index for each adjacent object identifier. Both sets of information (i.e. vegetation indexes for the pixels of the object itself and mean vegetation index of the adjacent objects) are included either by using the corresponding statistical features (mean and standard deviation) or the corresponding histograms.
3. Shape information: We include three basic features concerning the shape of the object. These characteristics are: the number of pixels that compose the object, the maximum number of pixels in a row, and the maximum number in a column.

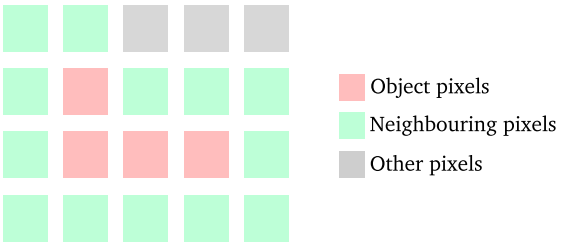


Fig. 2. Selection of adjacent pixels for one of the objects extracted

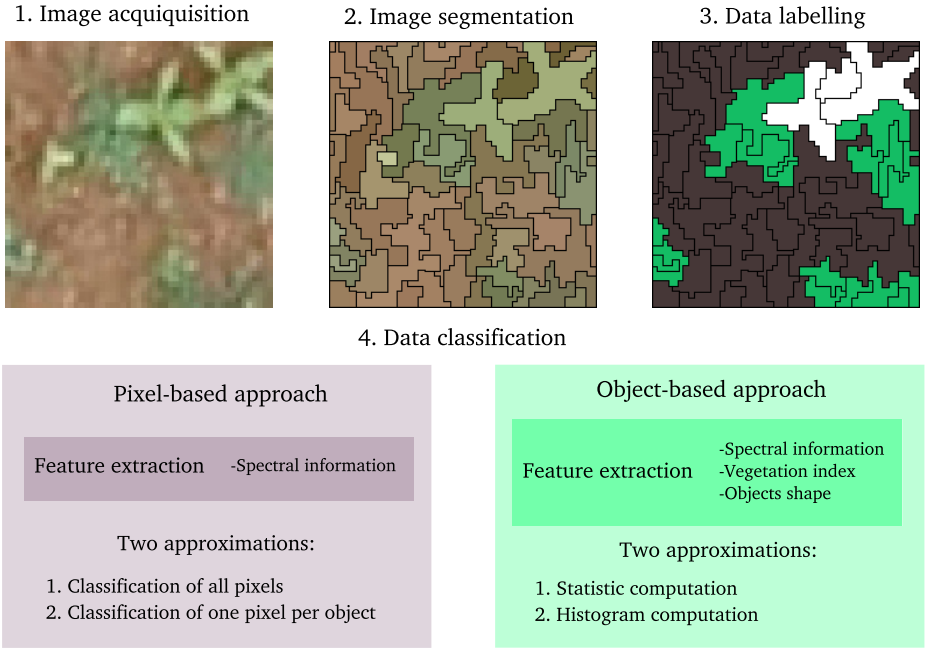


Fig. 3. Different stages for the tested methods

Fig. 3 shows the different stages of the method proposed and the different options tested.

4 Experimental Results

As previously introduced, three different cases are evaluated in this paper: Firstly, the use of pixels as basic units for classification, secondly, the use of objects, but not using the histogram information (only the mean and standard deviation), and, finally, the use of objects using the histograms.

As said, the segmentation algorithm presents two parameters (the number of clusters k and the minimum object size). These parameters have been optimised using the Johnson and Xie method [16] for measuring segmentation quality using the following values for each parameter: $\{50, 100, 200\}$.

We evaluate all the proposals by considering the one-vs-one multiclass version of the Support Vector Classifier (SVC) [18] given the proved good performance of SVC for remote sensing applications [19]. For the case of classifying all pixels of the objects, we used the linear kernel for SVC given the large amount of patterns (around $15,000$ training patterns, which makes very complicate the use of the Gaussian kernel in this case). For all nonlinear methods, the standard Gaussian kernel was used. In this case, the kernel width σ was selected within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, as well as the cost parameter of SVC, by means of a nested 5-fold cross-validation procedure applied to the training set.

To divide the data into different partitions and provide more robust results, a stratified 10-fold technique was performed, where the results were taken as the mean and standard deviation of the selected measures over the 10 test sets.

The results have been reported in terms of three metrics for measuring the performance of a classifier: 1) the well-known accuracy, *Acc*, which corresponds to the correct classification rate, 2) the Minimum Sensitivity ($MS = \min\{S_1, \ldots, S_k\}$), where S_i is the sensitivity for the class $\mathcal{C}_i$, and 3) the computational time of the classification method in seconds (training and testing using the Octave software). The measure considered during hyperparameter selection was *Acc*.

4.1 Results

The results of the experiments are shown in Table 1. Different conclusions can be drawn from this table. Firstly, concerning the pixel-based approach, the random selection of a pixel per object presents a poorer performance, probably because the selected pixel does not represent enough information about the object. Moreover, it can be seen that a simple pixel approach leads to reasonable performance (even when a linear algorithm is being used), although the computational cost is very high. Note that, the time recorded is that needed to train and test the model using a small portion of the experimental field. Therefore, the increase of 5 seconds to 1100 seconds is very important and could result in an intractable algorithm. However, in this case, we can appreciate the superiority in terms of performance of the object-based approach over the pixel-based one for all the metrics selected. In this sense, it can be seen that the use of statistical metrics is sufficient for obtaining very competitive results (being these results better than the one obtained by the use of histograms). Finally, the combination of the different sets of features improves the results to a great extent and results in a very promising classification of the data, while still maintaining a low computational cost.

Table 1. Mean *Acc*, *MS* and time results obtained in the test sets for the different settings and sets of input features considered

Dataset	Characteristics	*Acc*	*MS*	Time
Pixel-based analysis				
One pixel per object	Spectral	87.33 ± 8.29	76.00 ± 14.30	$\mathbf{4.53 \pm 0.02}$
All pixels	Spectral	90.89 ± 0.77	84.56 ± 1.76	1122.01 ± 299.37
Object-based analysis				
Statistical metrics	Spectral	93.33 ± 3.51	86.00 ± 5.16	4.66 ± 0.04
Statistical metrics	Spectral + VI	*94.33 ± 3.87*	*89.00 ± 5.68*	4.86 ± 0.02
Statistical metrics	Spectral + VI + Shape	$\mathbf{96.33 \pm 3.67}$	$\mathbf{93.00 \pm 6.75}$	5.03 ± 0.06
Histograms	Spectral	92.67 ± 5.62	87.00 ± 8.23	6.73 ± 0.04
Histograms	Spectral + VI	93.00 ± 5.54	85.00 ± 11.79	6.98 ± 0.04
Histograms	Spectral + VI + Shape	92.67 ± 5.62	85.00 ± 12.69	6.90 ± 0.06

The best results are highlighted in bold face and the second ones in italics.

In relation to the number of patterns used for training the model, it can be seen that the use of only 90 patterns per class (corresponding to the 9 training folds of the 10-fold experimental design) seems to be sufficient in this case. Nonetheless, it would be interesting to analyse how the model would change when providing a lower number of training patterns. We performed an additional experiment using a 2-fold stratified technique, instead of the previously used 10-fold approach (implying 50 patterns per class for training the model, instead of 90). The results for Acc and MS for the method based on the computation of the statistical metrics using the three sets of information are the following: 94.33 ± 0.47 in terms of Acc and 92.00 ± 0.00 in terms of MS. This is representative of the fact that the model can be trained accurately without much training information.

Fig. 4 shows the results obtained for the best performing method for a region of the experimental field. From this figure, it can be seen that the method performs reasonably identifying most crop and weed pixels. However, the methodology proposed classifies some soil pixels as crops (pixels that could be associated to imperfections on the ground such as rocks). This motivates the use of a more intelligent and thorough technique for selecting the objects to be labelled (e.g. objects that significantly differ in the spectral histogram from the rest).

Fig. 4. Representation of the different steps for the procedure considered: image acquisition, segmentation of the image, and, finally, classification

5 Conclusions and Future Work

This paper explores different classification alternatives to the problem of weed detection via Unmanned Aerial Vehicles in sunflower crops. More specifically, we have compared two common approaches (object-based analysis versus pixel-based analysis) and studied the influence of different data features and training samples in the performance of the classifier.

There are several conclusions that can be extracted from this work. It can be seen that an object-based analysis is more suitable than a pixel-based approach for the application considered. Firstly, because of the better performance, and

secondly because of the decrease of the computational time. Moreover, the use of other features, such as the information about the vegetation indexes or the shape of the objects, is meaningful and leads to better results. It has also been seen that the use of a small training sample (in this case, 90 patterns per class) is sufficient to yield a reasonable performance. However, there are still some issues that prevail and that will be considered in future studies. Firstly, it could be interesting to analyse which statistical metrics are the best to provide enough information for properly identifying weeds. Another issue is which data features are the best combination for this specific problem (e.g. texture features have not been studied in this case). Furthermore, it could be very useful to analyse if a specific technique for histogram classification (such as the histogram intersection kernel [12]) leads to better results. Finally, a method for selecting the most dissimilar patterns could be proposed, in order to be able to create more robust training samples.

References

1. European Crop Protection Association (ECPA): Url: http://www.ecpa.eu/information-page/industry-statistics-ecpa-total, industry statistics (2015)
2. Regulation (EC) 1107/2009 and Directive 2009128/EC: (2009)
3. Horizon 2020: Url: http://ec.europa.eu/programmes/horizon2020/
4. de Castro, A.I., Jurado-Expósito, M., Peña-Barragán, J.M., López-Granados, F.: Airborne multi-spectral imagery for mapping cruciferous weeds in cereal and legume crops. Precision Agriculture **13**(3), 302–321 (2012)
5. Castillejo-González, I.L., Peña, J.M., Jurado-Expósito, M., Mesas-Carrascosa, F.J., López-Granados, F.: Evaluation of pixel-and object-based approaches for mapping wild oat (avena sterilis) weed patches in wheat fields using quickbird imagery for site-specific management. European Journal of Agronomy **59**, 57–66 (2014)
6. Moranduzzo, T., Melgani, F.: Automatic car counting method for unmanned aerial vehicle images. IEEE Transactions on Geoscience and Remote Sensing **52**(3), 1635–1647 (2014)
7. Lucieer, A., Turner, D., King, D.H., Robinson, S.A.: Using an unmanned aerial vehicle (uav) to capture micro-topography of antarctic moss beds. International Journal of Applied Earth Observation and Geoinformation **27**, 53–62 (2014)
8. Peña, J.M., Torres-Sánchez, J., de Castro, A.I., Kelly, M., López-Granados, F.: Weed mapping in early-season maize fields using object-based analysis of unmanned aerial vehicle (uav) images. PLoS One **8**(10), e77151 (2013)
9. Torres-Sánchez, J., López-Granados, F., De Castro, A.I., Peña, J.M.: Configuration and specifications of an unmanned aerial vehicle (uav) for early site specific weed management. PLoS One **8**(3), e58210 (2013)
10. Torres-Sánchez, J., Peña, J., de Castro, A., López-Granados, F.: Multi-temporal mapping of the vegetation fraction in early-season wheat fields using images from uav. Computers and Electronics in Agriculture **103**, 104–113 (2014)
11. Blaschke, T.: Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing **65**(1), 2–16 (2010)
12. Barla, A., Odone, F., Verri, A.: Histogram intersection kernel for image classification. In: Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429). vol. 3, IEEE (2003). III-513-16

13. Meier, U., ed.: Growth stages of mono-and dicotyledonous plants, 2 edn
14. Bunting, P., Clewley, D., Lucas, R.M., Gillingham, S.: The remote sensing and gis software library (rsgislib). Computers and Geosciences **62**, 216–226 (2014)
15. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters **31**(8), 651–666 (2010)
16. Johnson, B., Xie, Z.: Unsupervised image segmentation evaluation and refinement using a multi-scale approach. ISPRS Journal of Photogrammetry and Remote Sensing **66**, 473–483 (2011)
17. Woebbecke, D.M., Meyer, G.E., Von Bargen, K., Mortensen, D.A.: Color indices for weed identification under various soil, residue, and lighting conditions. Transactions of the ASAE **38**(1), 259–269 (1995)
18. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks **13**(2), 415–425 (2002)
19. Mountrakis, G., Im, J., Ogole, C.: Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing **66**(3), 247–259 (2011)

A Novel Framework for Hyperemia Grading Based on Artificial Neural Networks

Luisa Sánchez[1(✉)], Noelia Barreira[1], Hugo Pena-Verdeal[2], and Eva Yebra-Pimentel[2]

[1] VARPA Group, Departament of Computer Science, University of A Coruña, A Coruña, Spain
luisa.brea@udc.es
[2] Optometry Group, Department of Applied Physics, University of Santiago de Compostela, Santiago de Compostela, Spain

Abstract. A common symptom of several pathologies is hyperemia, that occurs when a certain tissue has an abnormal hue of red. An increase of blood flow causes the engorgement of blood vessels, which produces the coloration. Hyperemia is an important parameter that specialists take into account when diagnosing diseases such as dry eye syndrome or problems derived from contact lenses wearing. In this work, we propose an automatic methodology to measure the hyperemia level of the bulbar conjunctiva. This methodology emphasizes the transformation from the extracted features to grading scales, using artificial neural networks for the process.

1 Introduction

Hyperemia, also known as erythema, occurs when there is an increase of blood flood in a certain tissue. The vessels engorge and the accumulation of blood causes an abnormal red coloration in the area. One of the areas where this symptom appears is the bulbar conjunctiva. Figure 1 shows different levels of hyperemia in the conjunctiva.

The hyperemia level of the bulbar conjunctiva is a symptom commonly used in diagnosis, as it is an early indicator of some pathologies, like the dry eye syndrome [10]. This pathology has a great incidence, specially in developed areas, since it is associated with computer use, the exposure to certain chemicals and some environmental factors such as pollution or dryness caused by air conditioners. It is also related with contact lenses wearing [13].

In order to measure hyperemia, specialists have to face a long and tedious process. First, they record a video of the eye to provide a view of the conjunctiva. Then, they have to select the best frame of the video for hyperemia grading, considering brightness and blurriness. This frame is analyzed in order to determine the hyperemia score. This score is not a value, but a level in a certain grading scale. Most commonly used scales are Efron and CCLRU [4]. Both of them are similar, as they present a set of images, each one of them serving as a prototype for each level. On one hand, Efron scale consists of five levels of severity, ranging

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 263–275, 2015.
DOI: 10.1007/978-3-319-19258-1_23

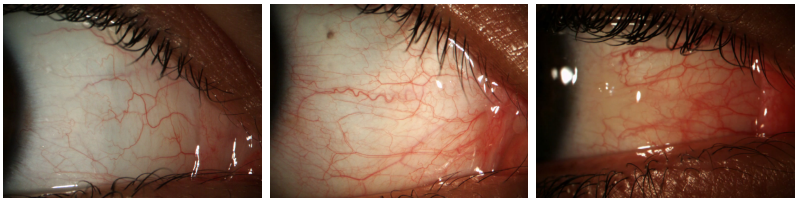


Fig. 1. Example of increasing hyperemia levels, from left to right

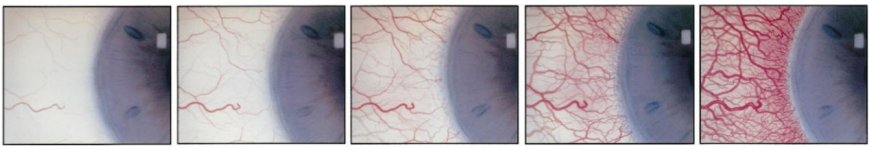


Fig. 2. Efron scale for conjunctival redness grading. From left to right, the levels are: normal, trace, mild, moderate, and severe.

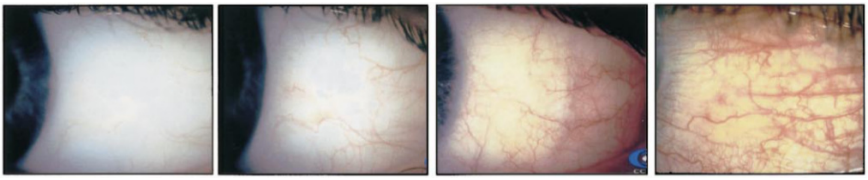


Fig. 3. CCLRU scale for bulbar redness grading. From left to right, the levels are: very slight, slight, moderate, and severe.

from 0 to 4, where 0 is the lower level, as depicted in Figure 2. On the other hand, CCLRU has only four levels, from 1 to 4. Also, it uses photographs in order to represent the prototypes, as it is shown in Figure 3.

Specialists compare the video frame with each image of the chosen scale in order to determine which one is the most similar. This is a complex process, as they have to analyze several indicators such as the quantity of vessels and the hue of the conjunctiva. Also, it is a subjective process that requires expertise, so that it presents a high intra and inter-expert variability.

Even if the scales have only four or five levels, it is common that the specialist concludes that the frame is between two categories, which leads him/her to assign a fractional value. Some studies [1,11] had tried to solve this problem with the development of grading scales consisting in a wider range of values, but they concluded that, though thinner scales produced more adjusted results, specialists were more confident with their gradings using only a reduced set of values. Another problem is that the highest and lowest values of the scales are very unlikely to happen so, in practice, specialists rate between the second lowest level and the second highest level instead of the real boundaries of the scale. Moreover, the reference grading scales are focused in different conjunctive areas and parameters.

There are several works in the matter of analysis of features for hyperemia calculation. Papas *et al.* [6] proposed several measures based both on color and vessel quantity or shape. Also, Park *et al.* [7] proposed other four measures based on several parameters, such as the number of vessels or the image contrast. The main limitation of these works is that none of them is totally automatic, as they require human intervention in order to select the region of interest for the analysis. Moreover, these works are focused in the development of measures for hyperemia grading and finding their correlation with the expert assessment, but they do not tackle the conversion from the computed feature values to a concrete grading scale. Also, these works analyze each feature separately, but it is suggested by specialists that several factors are taken into account during hyperemia grading.

In this work, we develop an automatic methodology for hyperemia grading according to Efron and CCLRU scales. The methodology computes the hyperemia value using a video as input. The frame most suitable for grading is selected. Then, a region of interest is automatically delimited, removing lids and lashes. Once this region is selected, a set of features are computed and, finally, these features are transformed to the desired scale by means of an artificial neural network.

This paper is structured as follows: Section 2 will explain the developed methodology. Then, Section 3 will depict the obtained results and, finally, Section 4 will detail our conclusions and future work.

2 Methodology

The input of the methodology is a video, about 20 seconds long, that shows the eye of the patient. The most suitable frame for grading is then selected, that is, the frame where it is easier for the specialist to appreciate the different features of the conjunctiva. Nevertheless, there is a lot of information in this frame that is not relevant for hyperemia grading, such as eyelids and eyelashes. This information must be removed since it could alter the results, so we define a region of interest that covers most of the conjunctiva. Next, several features are measured in the defined region. Lastly, these features are the input of different artificial neural networks, which perform the grading procedure.

2.1 Frame Selection

The first step of the methodology is the extraction of the most suitable frame of the video for grading. This operation has to keep in mind that the frame must provide a clear view of the eye features. To this end, the lightness of the image is measured by averaging all the image pixels in a specific color channel. The frame that presents the maximum average value is selected as candidate frame. Several measures were tested in order to find the best one: RGB luminance, HSV V-channel, HSL L-channel, and L*a*b* L-channel.

Moreover, the blurriness is also computed, as blurry frames can provide fake information for the grading. In order to compute this parameter, the Tenenbaum gradient algorithm was used [12]. The formula is depicted in Eq. 1, where S_x and S_y are the Sobel derivatives in each direction.

$$F_{Tenengrad} = \sum_{Height} \sum_{Width} S_x(x,y)^2 + S_y(x,y)^2 \quad (1)$$

2.2 Conjunctiva Extraction

The next step is to delimit the bulbar conjunctiva, so outer parts such as eyelids, lashes, iris and pupil must be removed. In order to delimit this region, we use an edge detection algorithm [3] and several morphological operations. This way, the weakest edges are removed and the conjunctiva contours are emphasized as Figure 4 shows. Then, we measure the distances from the left margin and the right margin of the image to the contour in each row. Since the iris curve is smoother than the corner of the eye, its vector of row distances is more uniform, so that we can identify both sides by analyzing these values. After that, we use a series of heuristics in order to select the center of the pupil and the corner of the eye.

For the pupil, we search for the two lowest non-zero values in the vector of distances corresponding to that side of the image (left side in Figure 4), one in the upper half of the image and another in the bottom half. These two points, depicted in red in Figure 4 delimit the pupil area, allowing us to find the center by obtaining its middle point.

For the corner of the eye, we tackle a similar process. We search the vector of distances corresponding to the side of the image where the corner of the eye is (in Figure 4, the right side), and look for the lowest non-zero distance in the central rows of the image.

Once we have the two reference points located, we use them to define a line and compute the middle point, that will serve as a center for an ellipse-shaped mask, $M_{ellipse}$ that comprises most of the conjunctiva, as it is depicted in Figure 5 (left). The minor axis of the ellipse is selected as the largest value that fits inside the conjunctiva contours. Also, we compute the angle between an horizontal line and the line defined by the two reference points (the horizontal axis of the ellipse) in order to know the rotation angle.

This procedure removes the eyelids in most cases, but we still have to face two additional problems: the lashes and some shiny areas that appear due to the camera light sources. In order to solve the former, we apply a binary threshold T_{lashes} to the original image, obtaining a new image, M_{lashes}, with the lashes (among other areas that are not relevant) colored in black (Figure 5, center).

In order to get ride of the shine, another binary threshold, T_{shine}, was used. This threshold is defined as a lower value than T_{lashes} because, in this case, we are searching for the brightest areas instead of the darkest ones. The thresholded image is then inverted in order to obtain another suitable mask (M_{shine}), as

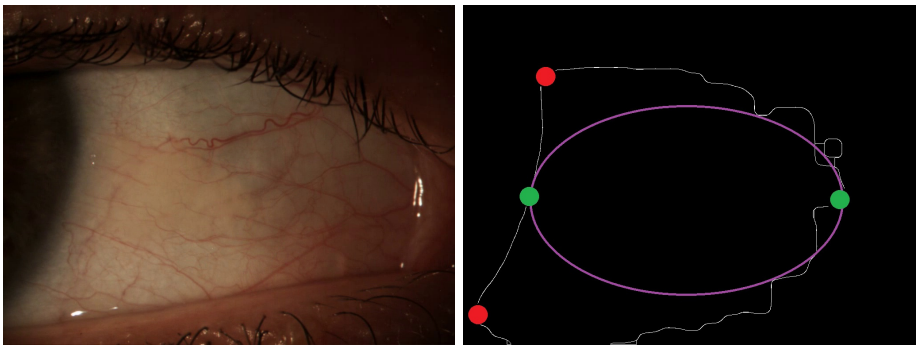


Fig. 4. Result of the application of the edge detection algorithm and morphological operations. The pupil center and the corner of the eye are highlighted in green, while pupil extremes are in red.

Fig. 5. Ellipse-shaped mask ($M_{ellipse}$), thresholded image in order to get rid of the lashes (M_{lashes}) and thresholded image in order to get rid of the shine (M_{shine})

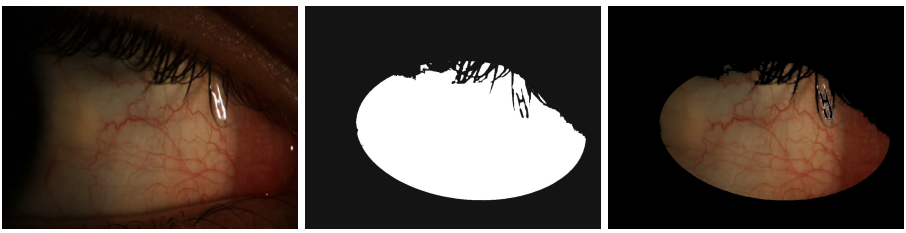


Fig. 6. Original frame, combined mask, and region of interest extracted using the masks

depicted in Figure 5 (right). $M_{ellipse}$, M_{lashes}, and M_{shine} are combined by means of an *and* operation.

With this approach, we obtain the mask that delimits the region of interest (Figure 6). Note that this mask removes some of the conjunctiva area but the remaining region contains the relevant analysis areas.

2.3 Hyperemia Features

In order to obtain the hyperemia level of an image, experts take into account several factors, such as the presence of vessels or the redness of the image. In this work, we extract different features from the image based both in color and quantity of vessels in order to perform the automatic grading. Several features were proposed in earlier works on the subject [6]. The features we had selected are depicted in Table 1. In these formulas, I is the input image; the variables R, G, B and H reference the image I channels, in the RGB and HSV colorspaces; E is the edge image computed by Canny algorithm, as depicted in Figure 7; n and m are the rows and columns of the image; i and j are the indices for the analyzed pixel; V represents all the edge pixels inside the conjunctiva mask, and M represents a mask defined as:

$$M_{ij} = \begin{cases} 0 & i \bmod step \neq 0 \\ 1 & i \bmod step = 0 \end{cases}$$

In our case, *step* is set to 10. This way, M does not cover the entire conjunctiva but a set of n_r image rows (10 in the experiments) that start from a reference row.

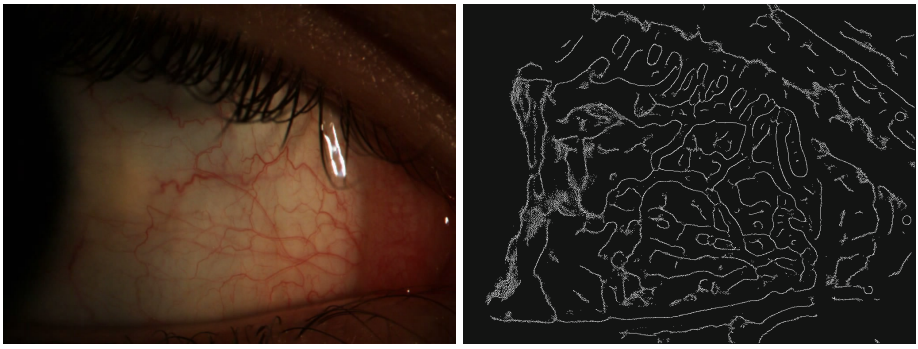


Fig. 7. Result obtained with edge extraction algorithm application

2.4 Hyperemia Grading

Once the hyperemia features have been computed, the result is a set of values, one for each metric and image. We want to transform the different ranges of values obtained by each feature into the range of the corresponding metric. In order to do so, we train an artificial neural network that, when receiving a certain input (a feature or a set of features) returns a grading value similar to the experts. We used two different approaches: a radial basis function (RBF) network [8,9] and a multi-layer perceptron (MLP) [2,5].

The selection of the networks was due to the characteristics of the problem and the data. On one hand, specialists first compare each image with the scale

Table 1. Implemented hyperemia features

Feature	Name	Formula
F1	Vessel count	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} E_{ij} \times M_{ij}}{n_r}$
F2	Vessel occupied area	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} V_{ij}}{n \times m}$
F3	Relative vessel redness	$\sum_{i=1}^{n}\sum_{j=1}^{m}(\frac{R_{ij} \times V_{ij}}{R_{ij}+G_{ij}+B_{ij}})$
F4	Relative image redness	$\sum_{i=1}^{n}\sum_{j=1}^{m}(\frac{R_{ij}}{R_{ij}+G_{ij}+B_{ij}})$
F5	Difference red-green in vessels	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}((R_{ij}-G_{ij}) \times V_{ij})}{n \times m}$
F6	Difference red-green of the image	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(R_{ij}-G_{ij})}{n \times m}$
F7	Difference red-blue in vessels	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}((R_{ij}-B_{ij}) \times V_{ij})}{n \times m}$
F8	Difference red-blue of the image	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(R_{ij}-B_{ij})}{n \times m}$
F9	Red hue value	$\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\lvert 128-H_{ij}\rvert}{n \times m}$

representative images in order to set the grade level. After that, since there is not a complete correspondence between the scale and the analyzed images, the expert decides a decimal part based on the distance of the image to the scale image. RBF networks have a similar behaviour, as its output depends on the distance of the input to each target. On the other hand, MLP is a model vastly used to solve complex problems with non-linear functions so it should also provide a good approximation.

3 Results

We performed experiments on 58 videos of bulbar conjunctiva. The videos were from both eyes and both sides of the eye (from the lacrimal to the pupil and from the pupil to the corner of the eye). These videos were captured by the Optometry Group of the University of Santiago de Compostela. Video resolution is 1024×768 pixels.

An expert graded the image set, using the two scales. He performed two gradings, months apart. This allowed us to validate the metric results. The intra-expert correlation is about 0.7 in both scales, as it is depicted in Figure 8. It can be appreciated how the coincident values do not form a thin line but a wider strip. This is because the measures have a variability of about ±0.5 points in the scale.

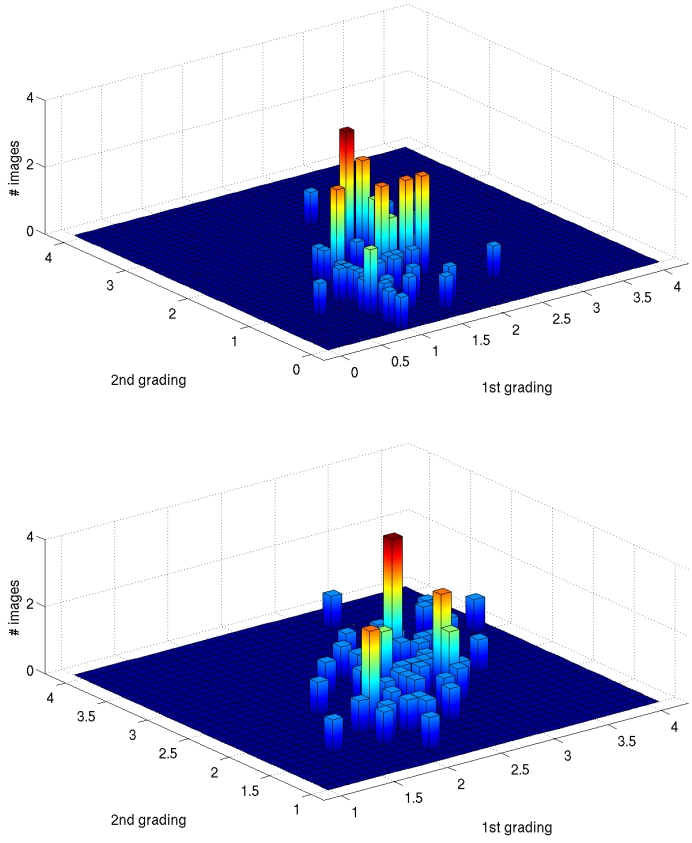


Fig. 8. Intra-expert correlation between the two different gradings. Top: Efron values. Bottom: CCLRU values.

The best frame for hyperemia grading was automatically extracted from each video. This step was validated by two experts with a success rate of 95% on average. This means that, in 95% of the cases, the extracted frame was labeled as the best one for the two specialists. The selected lightness measure was L channel of the L*a*b* color space.

We selected the best values for the thresholds on the ROI extraction stage. T_{lashes} is computed as the average gray level of the processed image because of the variability of lightness in the videos, that complicates the selection of a threshold value that works as desired for every image. T_{shine} was empirically set to 180 because the bright spots present a similar coloration in all the images, so a single value can be used through all the dataset.

We computed all the features for each image. Then, we compared each feature with the values from the specialist (mean of the two gradings) in each scale by means of the two proposed networks.

Both experiments were implemented in Matlab[1] R2014b with the neural networks toolbox 8.2.1. RBF networks have a spread ranging from 0.2 to 0.5. This value has been empirically determined for each network. RBF networks consist of two layers, a radial basis layer and an output layer. In the training stage, the radial basis layer contains zero neurons in the beginning and a new neuron is added to the network every step the stop condition is not met until the size of the layer reaches the size of the inputs. The MLP has a hidden layer with 10 neurons and uses a hyperbolic tangent sigmoid transfer function for hidden layers as well as a linear transfer function for the output layer. The bayesian regularization function was selected as training function, the gradient descent with momentum as learning function, and the mean square error as performance function.

Table 2. Average MSE for each feature

	RBFN		MLP	
Feature	Efron	CCLRU	Efron	CCLRU
F1	0.27553494	0.20167978	0.21706634	0.17716224
F2	0.46355412	0.77008845	0.21687044	0.21656307
F3	0.36794231	0.23370993	0.21688790	0.21913878
F4	0.25709127	0.23802481	0.21563993	0.21209613
F5	0.42318667	0.25380361	0.21922054	0.21915707
F6	0.26538561	0.22636582	0.21775561	0.21896880
F7	0.20137734	0.57606926	0.21498098	0.14942032
F8	0.20472369	0.37385429	0.22276596	0.14127769
F9	**0.15382070**	**0.15256991**	0.14492348	0.13963621
Combined	0.20463549	0.21316013	**0.11904438**	**0.09930078**

In order to evaluate the results, a k-fold cross-validation was used with $k = 5$ because of the size of the dataset. The tests show that the results for the MLP are better for all the features, with mean square error of 0.22 in worst case and 0.14 in best one. RBFN results are more variable, ranging from low mean square error (0.15) in feature $F9$, to a high error in $F2$ with both scales or $F7$ with the CCLRU scale.

[1] www.mathworks.com/products/matlab/

We also tested the use of the combination of the 3 best features ($F4$, $F6$, and $F9$) to train the networks, obtaining the results depicted in Table 2.

In view of the results, the best feature is $F9$ (red hue value) for all tests. All features provided good results with the MLP, which is the best of the two proposed approaches. The combination of the three metrics provided also good error values, improving the results for the MLP and supporting the idea of several parameters needed in order to provide an accurate grading.

Figure 9 shows the correlation between the expert gradings and the best feature, the combination of $F4$, $F6$, and $F9$ for a MLP network. The system gradings were computed as the average output of all the k-fold networks, and the correlation is 0.8 on average for all four cases. If we compare the expert gradings with the outputs of our system (Figures 10 and 11) we can see that our methodology behaves like an expert, as the dispersion of the automatic gradings with respect to the expert's is similar to the dispersion found in the expert's gradings (Figure 8).

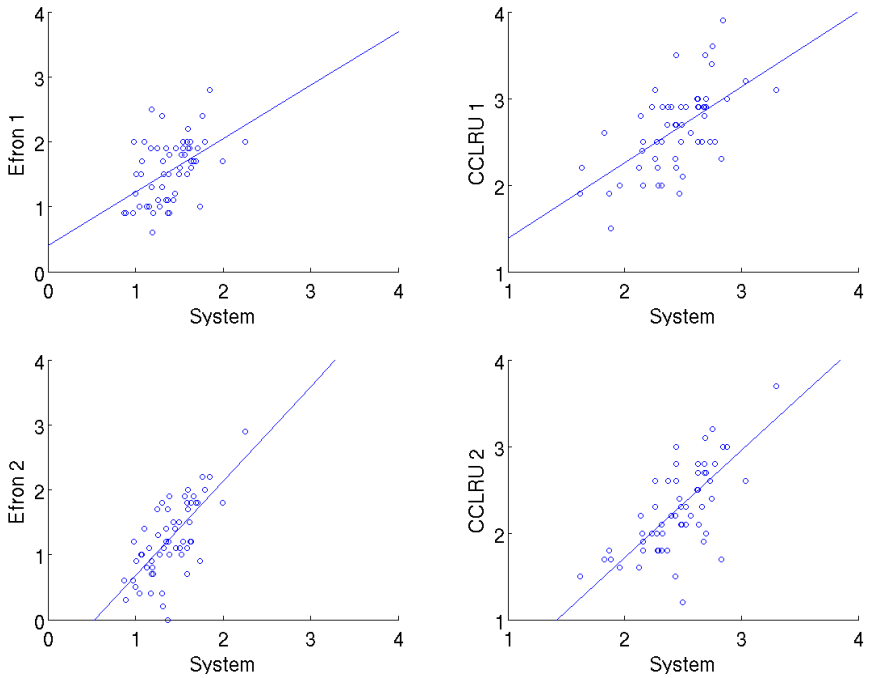


Fig. 9. Correlation between the first and second expert gradings and system outputs

4 Conclusions

In this paper, we propose an automatic methodology for hyperemia grading on bulbar conjunctiva videos. We first select the frame which is the most suitable for

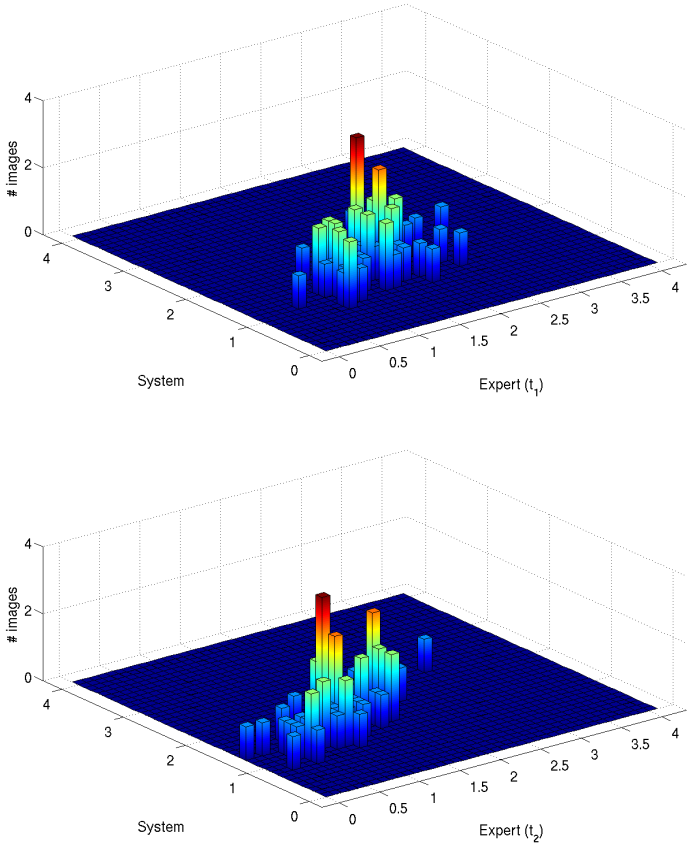


Fig. 10. Comparison between the first and second expert gradings and system outputs for Efron scale

grading from each hyperemia video. Then, we extract the region of interest, that comprises most of the conjunctiva. Once it is extracted, we compute the relevant features based on color and the presence of vessels. Lastly, we use artificial neural networks in order to perform the transformation from our scale to the specialists grading scale. The tests show that the MLP provides the best results with each feature. The results are also improved when several features are combined.

The methodology could be improved by means of refining the automatic ROI extraction in such manner that includes all the conjunctiva. Also, more experiments with new features could be conducted, both testing each feature separately and combining various features in different ways, in order to find the characteristics that are more relevant for the grading.

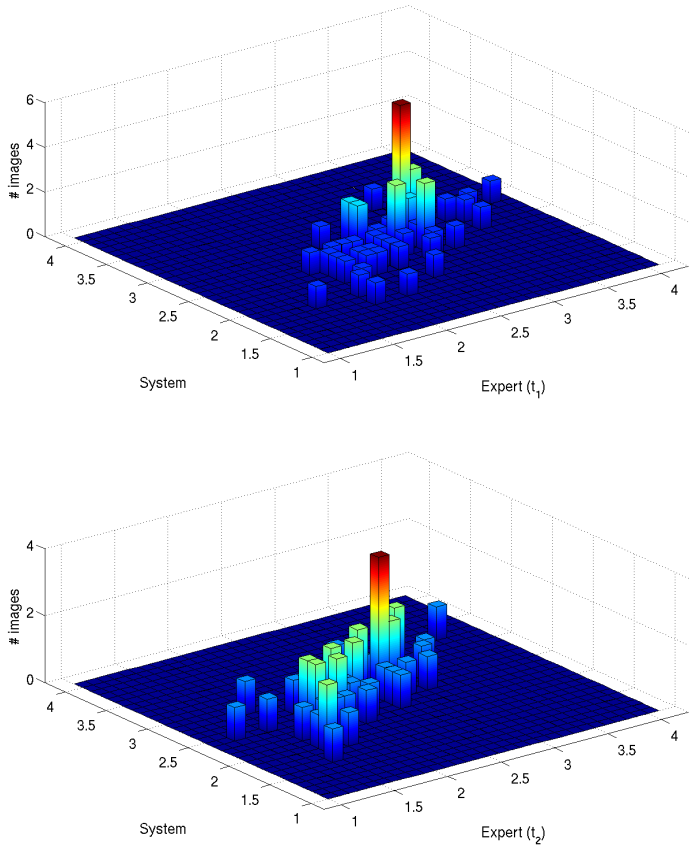


Fig. 11. Comparison between the first and second expert gradings and system outputs for CCLRU scale

References

1. Bailey, I., Bullimore, M., Raasch, T., Taylor, H.: Clinical grading and the effects of scaling. Investigative ophthalmology & visual science **32**(2), 422–432 (1991)
2. Baum, E.B.: On the capabilities of multilayer perceptrons. Journal of complexity **4**(3), 193–215 (1988)
3. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **6**, 679–698 (1986)
4. Efron, N., Morgan, P.B., Katsara, S.S.: Validation of grading scales for contact lens complications. Ophthalmic and Physiological Optics **21**(1), 17–29 (2001)
5. Gardner, M., Dorling, S.: Artificial neural networks (the multilayer perceptron)a review of applications in the atmospheric sciences. Atmospheric environment **32**(14), 2627–2636 (1998)
6. Papas, E.B.: Key factors in the subjective and objective assessment of conjunctival erythema. Investigative ophthalmology & visual science **41**(3), 687–691 (2000)

7. Park, I.K., Chun, Y.S., Kim, K.G., Yang, H.K., Hwang, J.M.: New clinical grading scales and objective measurement for conjunctival injection. Investigative ophthalmology & visual science **54**(8), 5249–5257 (2013)
8. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural computation **3**(2), 246–257 (1991)
9. Park, J., Sandberg, I.W.: Approximation and radial-basis-function networks. Neural computation **5**(2), 305–316 (1993)
10. Rolando, M., Zierhut, M.: The ocular surface and tear film and their dysfunction in dry eye disease. Survey of Ophthalmology 45, Supplement 2(0), S203–S210 (2001). http://www.sciencedirect.com/science/article/pii/S0039625700002034
11. Schulze, M.M., Jones, D.A., Simpson, T.L.: The development of validated bulbar redness grading scales. Optometry & Vision Science **84**(10), 976–983 (2007)
12. Sun, Y., Duthaler, S., Nelson, B.J.: Autofocusing algorithm selection in computer microscopy. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005), pp. 70–76. IEEE (2005)
13. Wolffsohn, J.S., Purslow, C.: Clinical monitoring of ocular physiology using digital image analysis. Contact Lens and Anterior Eye **26**(1), 27–35 (2003)

Applying a Genetic Algorithm Solution to Improve Compression of Wavelet Coefficient Sign

Antonio Martí[1], Otoniel López[2(✉)], Francisco Rodríguez-Ballester[1], and Manuel Malumbres[2]

[1] Universidad Politécnica de Valencia, Camino de Vera s/n, Valencia, Spain
[2] Universidad Miguel Hernández, Avda. Universidad s/n 03205, Elche, Spain
otoniel@umh.es

Abstract. Discrete Wavelet Transform has been widely used in image compression because it is able to compact frequency and spatial localization of image energy into a small fraction of coefficients. In recent years coefficient sign coding has been used to improve image compression. The potential correlation between the sign of a coefficient and the signs of its neighbors opens the possibility to use a sign predictor to improve the image compression process. However, this relationship is not uniform and constant for any image. Typically, image encoders use entropy coding to compact the wavelet coefficients information. This work analyzes the impact of the input parameters over a genetic algorithm that distributes into contexts (sets) the wavelet sign predictors in such a way that the overall aggregate entropy will be reduced as much as possible and a as a consequence higher compression rates can be achieved.

Keywords: Genetic Algorithm · Wavelets · Image coding · Sign coding

1 Introduction

Genetic algorithms were first introduced by Holland in [1] and they are nowadays well known and very popular to find optimal/suboptimal solutions in very large and complex problems [2].

In a genetic algorithm, the evolution usually starts from a population of randomly generated individuals and runs in generations. In each generation the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness value), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached.

In this work we are looking for an optimal/suboptimal solution of our particular problem related with wavelet image compressors. This kind of image compressor is based in the use of a mathematical transform called Discrete Wavelet

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 276–286, 2015.
DOI: 10.1007/978-3-319-19258-1_24

Transform (DWT). Wavelet transforms have proved to be very powerful tools for image compression, since many state-of-the-art image codecs, including the JPEG2000 standard [3], employ DWT into their algorithms. The energy of a wavelet transform coefficient is restricted to non-negative real numbers, but the coefficients themselves are not, and they are defined by both a magnitude and a sign. Shapiro stated in [4] that a transform coefficient is equally likely to be positive or negative and thus one bit should be used to encode the sign. In recent years, several authors have begun to use context modeling for wavelet sign coding [5–7], showing that despite the equiprobability of wavelet sign values, some sign correlation can be found among wavelet coefficients, resulting in overall compression ratio improvements.

In a previous work [8] we have observed that the sign of a wavelet coefficient may be strongly correlated with the sign of some neighbor coefficients. However, this relationship is not uniform and constant for any image, or even consistent within the same image. By increasing the number and kind of images under analysis, the relationship between the signs of the neighbor coefficients may be generalized or, on the other hand, it is possible that, when increasing the number of images, some relationships become contradictory.

In this work, after obtaining the sign prediction for a given neighborhood and its success probability, we will use a Genetic Algorithm (GA) to distribute all the neighborhood permutations into r sets in such a way that the overall aggregate entropy will be reduced as much as possible and a as a consequence higher compression rates can be achieved by the entropy encoder used in the wavelet-based image encoder. We will analyze the impact of the input parameters in the performance of the GA.

The paper is organized as follows: in section 2 we define the optimization problem. In section 3 we propose a GA algorithm that matches the problem definition and we perform a detailed analysis GA parameters that supply the best sign context distribution in the shortest time. The paper then presents the main results in section 4 and finally general conclusions are highlighted.

2 Context-Based Sign Coding Approach

Most wavelet image codecs do not consider the use of sign coding tools since the wavelet coefficients located at the high frequency subbands form a zero-mean process, and therefore they are equally likely positive as negative.

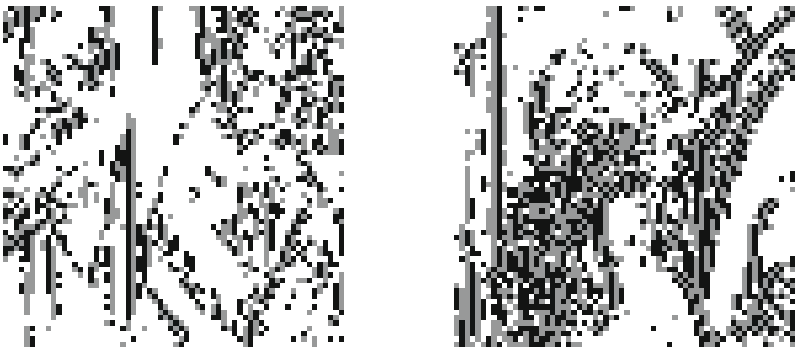
Schwartz, Zandi and Boliek [9] were the first authors to consider wavelet coefficient sign coding, using the sign of one neighboring pixel in their context modeling algorithm. The main idea behind this approach is to find correlations along and across edges.

The HL subbands of a multi-scale 2-D wavelet decomposition are formed from low-pass vertical filtering and high-pass horizontal filtering. The high-pass filtering detects vertical edges, thus the HL subbands mainly contain vertical edge information. Oppositely defined are the LH subbands that contain primarily horizontal edge information.

As Deever explained in [10], given a vertical edge in an HL subband, it is reasonable to expect that neighboring coefficients along the edge have the same sign as the coefficient being coded. This is because vertical correlation often remains very high along vertical edges in images. When a low-pass filter is applied along the image columns, it results in a series of similar rows, as elements in a row tend to be very similar to elements directly above or below due to the high vertical correlation. Subsequent high-pass filtering along similar rows is expected to yield vertically correlated transform coefficients.

It is also important to consider correlation across edges, being the nature of the correlation directly affected by the structure of the high pass filter. For Daubechies' 9/7 filters, wavelet coefficient signs are strongly negatively correlated across edges because this filter is very similar to a second derivative of a Gaussian as derived from theory of zero crossings and edge detection [11]. So, it is expected that wavelet coefficients will change sign as the edge is crossed. Although the discrete wavelet transform involves sub sampling, the sub sampled coefficients remain strongly negatively correlated across edges. In this manner, when a wavelet coefficient is optimally predicted as a function of its across-edge neighbors (e.g. left and right neighbors in HL subbands), the optimal prediction coefficients are negative, indicating an expected sign change. This conclusion is general for any wavelet with a shape similar to a second derivative of a Gaussian.

In Fig. 1 we plot the spatial distributions of signs in the HL subband of two popular test images: Barbara and Lena. The visible sign structures suggest that the sign bits of wavelet coefficients are compressible.



(a) Sign map for Barbara (b) Sign map for Lena

Fig. 1. Sign patterns in the HL subband for Barbara (left) and Lena (right). Black, gray and white dots for negative, positive and non-significant coefficients, respectively.

To estimate sign correlation in a practical way, we have applied a 6-level Dyadic Wavelet Transform decomposition of the source image and then a low quantization level to the resulting wavelet coefficients. Taking into account that the sign neighborhood correlation depends on the subband type (HL,LH,HH) as

Deever assesses in [7], we have used different neighbors in each subband type exploiting the correlation along and across the edges (see Table 1). So, for a particular subband type, we have defined n neighbors that can hold one of the three possible sign values, that is positive, negative and null (zero). This lead us to a set of 3^n different Neighbor Sign Patterns (NSP) for each subband type, being n the number of neighbors used in the sign prediction.

Table 1. Neighborhood analyzed

Neighborhood	HL subband **Neighbors**	LH subband **Neighbors**	HH subband **Neighbors**
3	N, NN, W	W, WW, N	N, W, NW
4	N, NN, W, WW	W, WW, N, NN	N, W, NW, NNWW
4b	N, NN, NNN, W	W, WW, WWW, N	N, W, NW, NNWW
5	N, NN, NNN, W, WW	W, WW, WWW, N, NN	N, W, NW, NNWW, NNNWWW

Other encoders like JPEG2000 and the one proposed by [12] use four neighbors (N,S,E,W) for the context formation, but since most non-embedded encoders use a Morton order (Z-order) [13] in the coding stage, no information is available about S and E neighbors and they cannot be employed in context formation. This represents a restriction when looking for sign correlation among the neighborhood which it is shared by most of the non-embedded encoders.

So, for each subband type and each neighborhood we can obtain the sign prediction table that contains the sign predictions, the number of occurrences and the probability of success for every $NSP\left[k\right], k = 1..3^n$. Then, when coding the sign of a wavelet coefficient in a particular subband, first we will get the sign value of the corresponding neighbor set in order to form the current NSP. Then we will look up this NSP in the table to find the sign prediction of the current wavelet coefficient. Finally, what we are going to encode is the correctness of this prediction. The performance of our binary entropy encoder will depend on the behavior of our sign predictor, the higher success prediction rate, the higher compression rate is achieved.

In order to improve the final compression performance of our entropy encoder, we propose the use of up to ten contexts for each subband type. So, for each subband type, we distribute the provided NSPs predictions into r sets (contexts) in such a way that the overall aggregate entropy will be reduced as much as possible and a as a consequence higher compression rates can be achieved. We have fixed to ten as a maximum the number of context in order not to increase to much the image encoder complexity, as in [6] where authors reduced and fixed the context number to five. We will use a GA algorithm to distribute all NSPs into r sets.

3 Genetic Algorithm for Wavelet Sign Prediction

The goal of our GA proposal will be to find a good context distribution of the NSPs prediction table for each subband type SB_t, where $t = \{HL, LH, HH\}$. So, for a particular wavelet coefficient $C_{i,j}$ with sign value $SC_{i,j} = \{+, -, NULL\}$, that belongs to subband SB_t, the prediction table will provide a sign prediction, $\hat{SC}_{i,j}$, based on its Neighborhood Sign Pattern, $NSP\,[k]$, where k represents the specific sign pattern of the neighborhood of the current wavelet coefficient, $C_{i,j}$ and the set (context) number where the NSP will be assigned for the entropy encoder.

There is no univocal relationship between a neighbor sign permutation and the sign prediction, i.e not always for a same $NSP\,[k]$ pattern, the sign prediction of the current wavelet coefficient, $\hat{SC}_{i,j}$, is always positive or negative. However, it is possible to find out that for a particular $NSP\,[k]$ sign pattern, $\hat{SC}_{i,j}$ is more probably to be positive than negative or vice versa. But, the problem is still more complex, because a sign prediction for a neighbor sign pattern could fit well for an image and not for others. Evenmore, the context distribution and the number of context to use will affect the encoder compression performance. Therefore, the idea is to find suboptimal neighbor sign pattern predictions and context distribution that better fit for a representative set of images, so we can capture the canonical wavelet sign redundancy introduced by a particular wavelet filter. In this manner, once the universal prediction table is found, it could be used at both encoder and decoder sides.

At this point, the motivation of using GAs to compress the sign of wavelet coefficients is twofold. First, when the number of selected neighbors for the analysis of sign correlation grows or when there is a great number of images to be used in the analysis and also when the number of context to distribute the NSPs in grows, the search space is excessively wide. Second, it is not intuitive to find a way of combining the predictions obtained for several images, and mix them in an unique prediction table.

The context distribution problem is similar to the mathematics problem of finding the ways to partition a set of k objects into r sets. The number of possible partitions is called Stirling number of the second kind [14] and is denoted by $S(k, r)$ (see equation 1). For example, if we use three neighbors for the sign prediction we have 27 NSPs (3^3) and distributing then into 5 contexts, the Stirling number of the second kind is 61338207158409090, which means that we have that number of possibilities to distribute the 27 NSPs into 5 contexts.

$$S(k,r) = \frac{1}{r!}\sum_{j=0}^{r}(-1)^{r-j}\binom{r}{j}j^k \tag{1}$$

4 Genetic Algorithm Design and Analysis

In this work, we analyze the effect of several parameters over the results (fitness and convergence) obtained by the GA. The parameters analyzed are:

- Population size
- Mutation probability
- Neighborhood
- Context number (sets)

We have executed the GA using the following variations. For each execution we have performed 100 iterations because of the influence of the random seed over the generation of the best individual. This lead us to a total of 100,000 executions.

- Population size: 10, 25, 50, 100, 150, 200, 300, 400
- Mutation probability: 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001 (from 5% to 0.1%).
- Neighborhood: 3, 4, 4b, 5
- Context number (sets): 4, 5, 6, 7, 8, 9

In Fig. 2 we show the normal probability graphic for the fitness values. As it can be seen, they are not normal and so we will use other kind of analysis, mainly based on dispersion graphics.

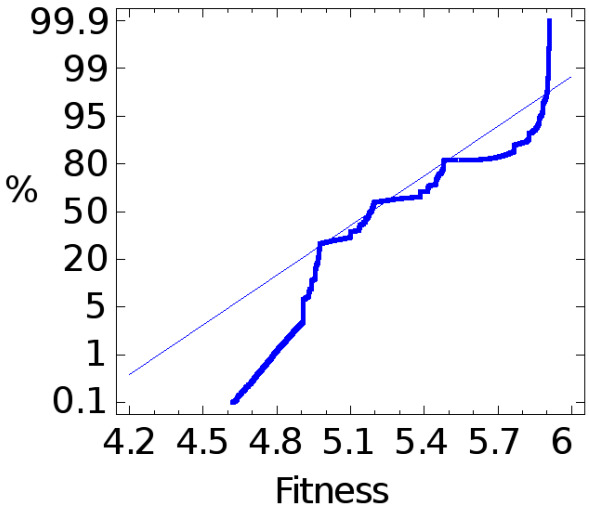


Fig. 2. Normal probability graphic

In Fig. 3 we show the best result obtained in each GA execution, where the results are ordered as a function of the GA parameter values used in the the following script.

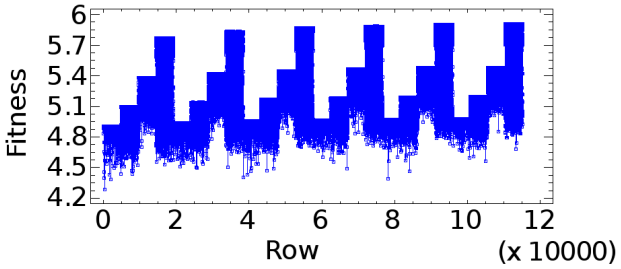


Fig. 3. Chronological sequence of GA fitness results

```
For context (from 4 to 9) do
 For neighbors in (3; 4; 4b; 5) do
  For population_size in (10; 25; 50; 100; 150; 200; 300; 400) do
   For mutation_probability in
     (0.05; 0.01; 0.005; 0.001; 0.0005; 0.0001) do
      Run GA #images.txt #contexts #neigbors
          #population_size #mutation_probability
```

where images.txt contains the wavelet coefficients sign prediction for each NSP and image set.

In Fig. 3 we can see 6 successive stairs. All stairs have a similar shape, but the results are different on each stair. The number of stairs correspond to the number of contexts analyzed, i.e 4, 5, 6, 7, 8, 9. Looking at that figure we can conclude that we obtain a better compression as the number of context used in the entropy encoder increases and also, that the rest of parameters has nearly no influence over context parameter.

In each stair, there are 4 steps which corresponds to the neighborhood analyzed, i.e 3, 4, 4b and 5. The effect of this parameter is quite significative and it is represented by the difference between steps. This difference can be seen on both the minimum and maximum fitness values. At each step or neighborhood we can appreciate an upper limit which is the maximum compression independently of the rest of GA parameters (population size and mutation probability). Obviously, this limit is higher for the neighborhood that uses 5 neighbors. Otherwise, there is a lower bound on each step. The lower bound for 3 and 4b neighborhood is quite similar, being this lower bound higher for the neighborhood 4. This effect is more clearly shown in Fig. 4 where fitness values are represented as a function of the neighborhood. At this point, we could not asses which neighborhood, 4b or 5 is the best one to be used in the compression algorithm because neighborhood 5 obtains better results but with a greater dispersion than neighborhood 4b. Moreover, compression algorithm complexity increases as the number of neighbors does.

The results variation in each neighborhood shown in Fig. 4 is due to the intrinsic GA parameters like population size and mutation probability. In order

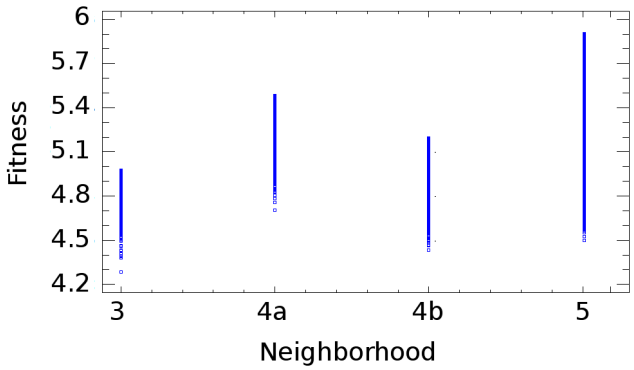
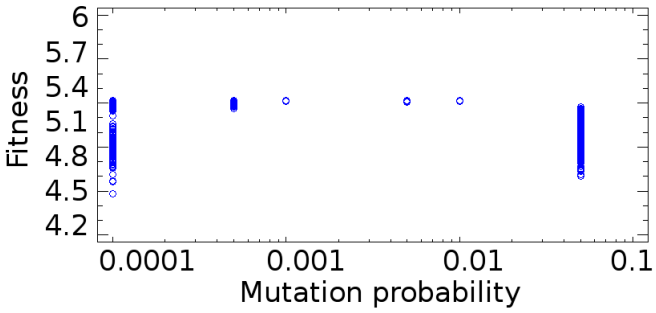
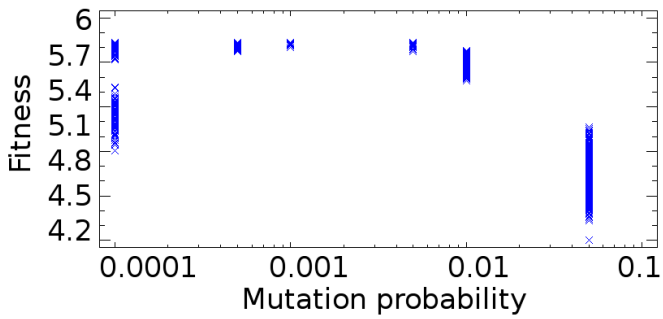


Fig. 4. Dispersion fitness vs neighborhood



(a)



(b)

Fig. 5. Dispersion fitness vs probability using neighborhoods a) 4b, b) 5

to easily analyze the results, we will reduce the analysis to the context number 6 and neighborhood 4b and 5. In Fig. 5 (a) and Fig. 5 (b) we can see the dispersion graphic for the fitness values as a function of the mutation probability for those

executions that used the context number 6. Fig. 5 (a) represents executions using 4b neighborhood and Fig. 5 (b) represents executions using 5 neighborhood. As it can be seen for the range from 0.0005 to 0.005, the mutation probability does not affect to the fitness values obtained by the GA. Using lower mutation probability it is possible to obtain suboptim results, but an important number of executions obtain very low fitness values. On the other hand, with mutation probabilities higher than 0.005 the algorithm fluctuates in a random way and the fitness results get worse considerably, both the maximum value and the mean value.

Analyzing the results when the mutation probability values are in the range from 0.0005 to 0.005, we can observe that results are grouped (gathered), i.e. there are less variability and also we can observe that we obtain better results when using neighborhood 5.

Finally we will analyze the effect of population size over the fitness results. In Fig. 6 we can see the dispersion graphic for all executions and as shown there is a great variability in the results, but this variability and also the range are quite similar for all population sizes except for population size of 10 that presents the worst results.

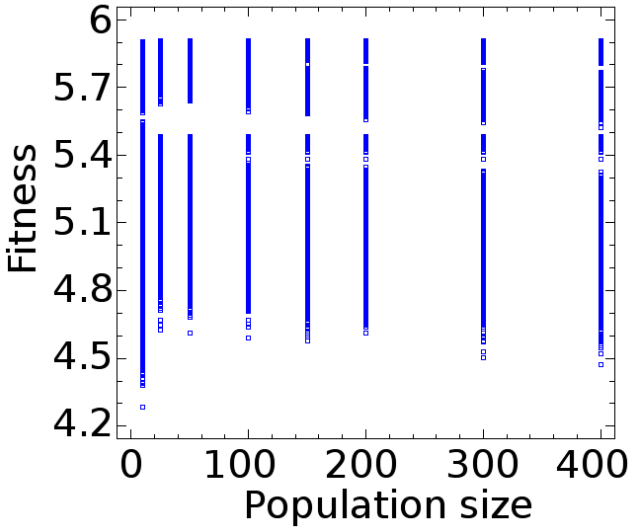


Fig. 6. Dispersion fitness vs population size

Previous analysis show that the best results are obtained for the neighborhood 5 and mutation probabilities from 0.0005 to 0.005. Fig. 7 shows the dispersion graphic for mutation probabilities in the range of 0.0004 to 0.006 using neighborhood 5 and 6 contexts. We can see three groups of results. First, with a population size of 10 individuals, the GA is not able to achieve as good results

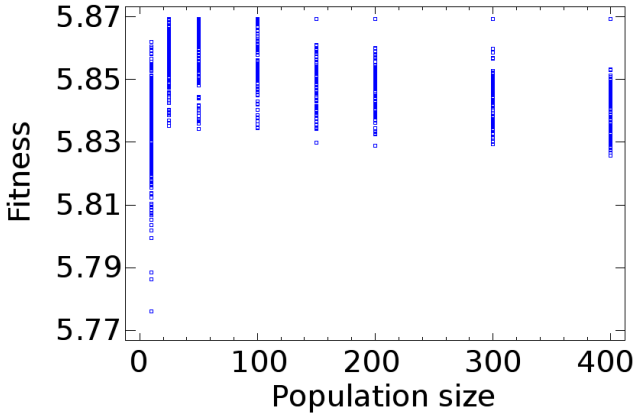


Fig. 7. Dispersion fitness vs population size for ($mutationprobability > 0.0004$) and ($mutationprobability < 0.006$) and ($neighborhood = 5$) and ($context = 6$))

as when using greater population sizes. Secondly, population sizes of 25, 50 and 100 individuals present optimal and identical results. Third, for populations sizes of 150, 200, 300 and 400 the graphic does not allow to see if the results are as good as for the previous group. In this last group, the GA algorithm achieve the optimum result, but in a great number of executions, the result is not the optimum and there is a greater dispersion.

Table 2. Execution results statistics for mutation probability from 0.0004 to 0.006, using neighborhood 5 and 6 contexts

Population	**25**	**50**	**100**	**150**	**200**	**300**	**400**
Average	5.85746	5.86096	5.86153	5.86004	5.86034	5.85782	5.85712
Standard Deviation	0.00671	0.00835	0.01034	0.01182	0.01181	0.01397	0.01517
Variation Coefficient	0.1146%	0.1426%	0.1765%	0.2017%	0.2016%	0.2385%	0.2591%
Minimum	5.83509	5.8343	5.83435	5.82985	5.8289	5.82945	5.82575
Maximum	5.86922	5.86922	5.86922	5.86922	5.86922	5.86922	5.86922
Range	0.03413	0.03492	0.03487	0.03937	0.04032	0.03977	0.04347

In order to determine if there are differences when using population sizes of 25, 50, 100, 150, 200, 300 and 400, we present Table 2 with mean, median and standard deviation values. As it can be seen, both standard deviation and variation coefficient grow as the population size does.

5 Conclusions

In this paper we have analyzed the input parameter of a GA that is able to distribute the sign prediction for a given neighborhood into r sets obtaining the

lower aggregate entropy. From the analysis performed, we can conclude that the algorithm obtains better results as the number of context used increase. Even more, the neighborhood used for the wavelet sign prediction has a greater impact over the results, being the neighborhood that uses 5 neighbors the one that better results obtains. Regarding intrinsic parameters of the GA, when we use mutation probabilities in the range from 0.0005 to 0.005 we obtain less variability in the execution results. Finally, we also have analyzed the population size, being those in the range from 25 to 100 the ones that better results and behavior obtain.

References

1. Holland, J.: Adaption in Natural and Artificial Systems. University of Michigan Press (1975)
2. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
3. ISO/IEC 15444–1: JPEG2000 image coding system (2000)
4. Shapiro, J.: A fast technique for identifying zerotrees in the EZW algorithm. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing **3**, 1455–1458 (1996)
5. Wu, X.: High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression. In: Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers, pp. 1378–1382 (1997)
6. Taubman, D.: High performance scalable image compression with EBCOT. IEEE Transactions on Image Processing **9**(7), 1158–1170 (2000)
7. Deever, A., Hemami, S.S.: What's your sign?: Efficient sign coding for embedded wavelet image coding. In: Proc. IEEE Data Compression Conf., Snowbird, UT, pp. 273–282 (2000)
8. Lopez, O., Martinez, M., Piñol, P., Malumbres, M., Oliver, J.: E-ltw: An enhanced ltw encoder with sign coding and precise rate control. In: 2009 16th IEEE International Conference on Image Processing (ICIP) pp. 2821–2824, Nov 2009
9. Schwartz, E.L., Z, A., Boliek, M.: CREW: Compression with reversible embedded wavelets. In: Proc SPIE, pp. 212–221 (1995)
10. Deever, A., Hemami, S.S.: Efficient sign coding and estimation of zero-quantized coefficients in embedded wavelet image codecs. IEEE Transactions on Image Processing **12**(4), 420–431 (2003)
11. Mallat, S., Zhong, S.: Characterization of signals from multiscale edges. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(7), 710–732 (1992)
12. Tian, C., Hemami, S.S.: An embedded image coding system baed on tarp filter with classification. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, Canada, May 2004
13. Morton, G.M.: A computer oriented geodetic data base and a new technique in file sequencing. Technical report, IBM Ltd (1966)
14. Riordan, J. In: Introduction to Combinatorial Analysis. Princeton University Press (1958)

Finding the Texture Features Characterizing the Most Homogeneous Texture Segment in the Image

Alexander Goltsev[1(✉)], Vladimir Gritsenko[1], Ernst Kussul[2], and Tatiana Baidyk[2]

[1] International Research and Training Centre of Informational Technologies and Systems of National Academy of Sciences of Ukraine, Kiev, Ukraine
agoltsev@adg.kiev.ua
[2] Centro de Ciencias Aplicadas y Desarrollo Tecnológico, Universidad Nacional Autónoma de México, Circuito Exterior S/N, Ciudad Universitaria, AP 70-186 CP 04510, México D. F., México
ekussul@unam.mx

Abstract. We propose an algorithm for finding a set of texture features characterizing the most homogeneous texture area of an input image. The found set of features is intended for extraction of this segment. The algorithm processes any input images in the absence of any preliminary information about the images and, accordingly, without any learning. The essence of the algorithm is as follows. The image is covered with a number of test windows. In each of them, a degree of texture homogeneity is measured. The test window with maximal degree of homogeneity is determined and a representative patch of pixels is detected. The texture features extracted from the detected representative patch is considered as those that best characterize the most homogeneous texture segment. So, the proposed algorithm facilitates solution of the texture segmentation task by providing a segmentation technique with helpful additional information about the analyzed image. A computer program simulating the algorithm has been created. The program is tested on natural grayscale images.

Keywords: Texture · Texture feature · Texture window · Texture segmentation · Homogeneous texture

1 Introduction

The problems of image segmentation, texture segmentation, and perceptual grouping are very similar. Actually, they are the same problem that belongs to a low-level vision task and that remains to be of a great importance in the field of computer vision (e.g., [1–6]).

Numerous papers devoted to the texture segmentation problem have been published to date. In the majority of them, statistical analysis of images is used for description, recognition and segmentation of textures. For example, a filter-based statistical approach for texture recognition is presented in [7, 8]. Adaptive-scale filtering is a variant of this approach [9], as well as a multi-scale aggregation of filter responses and shape elements [10], and hierarchical clustering of image segments and texture descriptors [11].

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 287–300, 2015.
DOI: 10.1007/978-3-319-19258-1_25

In its general formulation, the problem of image segmentation is very complicated. Therefore, often, the segmentation problem is facilitated by different ways. In particular, the segmentation technique may be provided with a set of training points belonging to those textures that should be recognized and segmented in the given images. In this case, the task apparently falls into the category of supervised learning so that the segmentation technique can tune its parameters by means of learning on those training points. During the segmentation process, the technique analyzes an input image and classifies all its pixels to some class from a given set of texture classes. The approach to texture recognition and segmentation based on supervised learning is presented in a significant number of publications (e.g., [13–15]). In particular, the algorithms based on neural networks that serve as classifiers are described in [16–24].

Another approach to the texture segmentation problem implies that the segmentation algorithm uses some universal texture features to extract texture areas without any learning. This approach falls into category of unsupervised image and texture segmentation or a data driven approach (e.g., [25–29]). Many unsupervised segmentation algorithms demonstrate good results, but most of them require knowledge about the number of texture segments present in the input image. In this approach input images should be processed in the absence of preliminary information about them. Many texture segmentation algorithms are proposed that do work in the absence of information about processed images (e.g., [12, 30]). For example, such an algorithm is described in [12], where texture is characterized by a probability density function on which statistical variations of texel properties depend.

Ideally, this approach should solve such high-level problems as extraction of all objects present in the image from a background, recognition of these objects by parts, determination of their relative positions, and, then, creation of complete description of the given scene. Clearly, solution of such a complex task should consist of several stages. As the lowest-level stage, we propose to delineate areas of approximately homogeneous fine-grained texture segments in the image. By the term “approximately homogeneous fine-grained texture segment” we understand the following. In general, images consist of distinct areas representing different surfaces of objects present in the given scene. Optically heterogeneous objects’ surfaces with discontinuities in depth, material, illumination, etc., usually correspond in the image to high-variability regions that are non-texture or coarse texture segments. These regions should not be extracted at the first stage of the image analysis. Instead, we propose to delineate the low-variability areas with fine granularity and with a relatively smooth and fine scale variation of image brightness and with a relatively regular pixel-level noise. These smooth fine-granularity-texture segments we consider as homogeneous.

After extraction of all homogeneous texture segments, the remainder of the image should consist of the boundaries between the segments, coarse texture areas, and strongly inhomogeneous texture regions. As a result of such image processing, the subsequent analysis of the image should become much easier. We have created a model for sequential extraction of such homogeneous texture segments. However, the whole model is rather complex and its explanation requires a long description. Therefore, the present paper describes only a part of the segmentation model, videlicet, an algorithm for finding the texture features characterizing the most homogeneous texture segment present (among others) in the input image. In this work, this texture area is, then, delineated by the segmentation model and presented in the figure to demonstrate which segment is chosen by the algorithm as most homogeneous.

2 General Description of a Texture Segmentation Model

The core ideas of the model for texture segmentation are as follows. First of all, a set of texture features is found that characterizes the most homogeneous segment present in the input image. This feature set is found together with the representative patch from which it is extracted (this is the model part we describe in the paper). To find the characterizing feature set and the representative patch, the whole image is covered with a number of test windows. In each test window a degree of texture homogeneity is measured. The test window with the maximal degree of homogeneity is determined and, then, the representative patch is detected within this window. The found representative patch is considered as belonging to the most homogeneous texture segment of the image and is used for its extraction as follows. The patch is assigned to be a core of the extracted texture segment and the segmentation procedure gradually extends the patch to the boundaries of the segment using the set of texture features extracted from the patch. As a result of this procedure, a single approximately homogeneous and simply connected fine-grained texture segment is extracted.

After extraction of the current segment is completed, its area is excluded from further consideration. At the next iteration, extraction of the next texture segment starts again from finding a new set of texture features and a new representative patch that characterize the new most homogeneous segment present in the remainder of the image.

Thus, according to above description, the algorithm for finding such a set of texture features that characterizes the most homogeneous texture area of the image solves some intermediate task within the complex image segmentation problem. The proposed algorithm may be useful for any segmentation techniques because it considerably reduces uncertainty of the analyzed image and facilitates solution of the whole segmentation problem by means of providing a segmentation technique with helpful additional information about the image. The algorithm processes all input images without any learning and without any preliminary information about them.

3 Description of Textures

A set of M texture features (feature types) is used for texture description. The texture features serve to measure similarity and difference between different image patches. In order to evaluate texture characteristics of image points, square texture windows of the same size are used. The whole input image is covered with a large number of overlapping texture windows, let us designate their number as N. The whole set of M texture features is extracted from every texture window. Hereinafter, an index m is applied for indexing texture features. All pixels of a window take part in computation of every texture feature. The whole set of all M texture features computed in a texture window is associated with its central pixel.

Let only images of the same size be presented for segmentation. In such a case, configuration of the whole set of N texture windows is constant with the same coordinates of their centers in the input image. All measured values of texture features are associated just with these centers of texture windows. Therefore, we introduce a continuous coordinate system of the window centers of $N = I \times J$ pixels with indices i and

j (i = 1, 2, 3, ..., I; j = 1 , 2, 3, ..., J), let us name this coordinate system as a C-grid. Hereinafter, we describe all processes, procedures and operations only in the coordinates of the C-grid.

Every feature value computed in a texture window is some analog value (a scalar). All these analog values are normalized to the same number U. All these M normalized values of texture features computed in a texture window constitute a feature pattern describing the corresponding texture window; this pattern is associated with a proper pixel of the C-grid. Let us introduce an integer three-dimensional array $\mathbf{F}[I][J][M]$ for representation of all N such feature patterns, so that each (i, j, m)-th element of the array $\mathbf{F}$ is the number between 0 and U which represents the m-th texture feature computed in the (i, j)-th texture window. The array $\mathbf{F}$ is the texture feature description of the whole input image.

4 Algorithm for Finding the Texture Feature Pattern Having the Maximal Number of Similar Patterns in a Local Image Patch

In order to evaluate the degrees of homogeneity of different parts of the image, a set of square test windows of the same size are applied, analogously to the system of texture windows introduced above. The whole input image is covered with a number of test windows; let us designate their total number by Z. The test window size is larger than that of a texture window. The test windows may have larger or smaller size, may overlap with one another or may not. This all depends on what degree of accuracy we wish to get and how much time we can spend on it.

Let us introduce a binary three-dimensional array $\mathbf{Q}[z][i][j]$ of the $Z \times I \times J$ size. The totality of one-valued elements of the array $\mathbf{Q}[z]$ represents texture windows contained in each z-th test window.

A degree of homogeneity of every image part is measured by the number of feature patterns similar to each other within the corresponding test window as follows. Let us introduce an integer tree-dimensional array $\mathbf{S}[z][i][j]$ of $Z \times I \times J$ elements. Each element $\mathbf{S}[z][i][j]$ serves to count the number of those feature patterns (extracted from all texture windows containing in the z-th test window) that are similar to the pattern of the (i, j)-th texture window. Thus, functionally, each element of the array $\mathbf{S}$ is simply a counter. All elements of the array $\mathbf{S}$ represent the texture homogeneity distribution in the whole image. Before formation of the array $\mathbf{S}$, all its elements are set zero-valued.

Let us consider a procedure of formation of the (z, i, j)-th element of the array $\mathbf{S}$ by example of comparison between the feature pattern of the (i, j)-th pixel (of the C-grid) and the feature pattern of some other (x, y)-th pixel. For every pair of such feature patterns, M comparisons (according to the total number M of the used texture feature types) are performed between all their corresponding (i.e., having the same index m) normalized feature values. For each of these M comparisons, the absolute difference between two feature values is calculated. This difference is compared with a predefined threshold H. If all M such differences lie inside the interval H, the feature patterns of the (i, j)-th and the (x, y)-th pixels are considered similar and, therefore, the

(i, j)-th element of the array $\mathbf{S}[z][i][j]$ (the (z, i, j)-th counter) is incremented by one. In all details, the algorithm for finding the representative pixel of the most homogeneous image segment is described by **Algorithm 1** in the pseudo-code format.

The designation $\mathbf{1}(\ )$ in **Algorithm 1** is the unit step function which is defined by the formula

$$\mathbf{1}(k) = \begin{cases} 1, & \text{for } k > 0, \\ 0, & \text{for } k \le 0. \end{cases}$$

Thus, the parameter H is the threshold which divides the feature patterns of the compared texture windows of the z-th test window into "similar" and "dissimilar" ones. The above description implies that the same threshold of similarity H is used for all M feature values. However, of course, different thresholds H_{m} may be applied for different feature types in the process of comparison between the corresponding feature values of two compared texture windows.

We suppose that the found representative pixel which corresponds to the maximal number of similar feature patterns is the pixel that belongs to the most homogeneous texture segment of the image and that its feature pattern is typical and distinctive for this texture segment. Therefore, we postulate that the feature pattern extracted from the texture window corresponding to the representative pixel adequately describes and characterizes this most homogeneous texture segment. In order to obtain larger set of feature patterns which, in the aggregate, will better characterize this segment, the representative pixel may be expanded to its nearest vicinity. And the feature patterns extracted from all texture windows corresponding to the pixels of this expanded representative patch may be used to form a more representative feature description of the sought-for texture segment. This description is formed by combining all the patch's feature patterns.

In the procedure of extraction of each homogeneous texture segment by the texture segmentation model, the found representative pixel is assigned to be the first and core pixel of the sought-for texture segment, i.e., it serves as an origin from which the model starts expanding the sought-for segment's area to its boundaries.

The parameter $S_{\max}$ which is computed according to **Algorithm 1** is rather informative. It is a characteristic (a degree) of homogeneity of the most homogeneous texture segment present in the whole image (or in its preliminarily defined region). During the process of sequential extraction of all homogeneous texture segments from the input image, this parameter enables the segmentation model to evaluate a degree of homogeneity of the next segment. This, in its turn, allows the segmentation model to refuse from its extraction if the value of $S_{\max}$ becomes too small. Let us designate the test window size as D. According to **Algorithm 1**, the maximum possible value of $S_{\max}$ is D^2. Then, if, for example, the $S_{\max}$ value computed for the next segment becomes less than $D^2/2$, this certainly indicates that the considered segment is not enough homogeneous. Since the segmentation model is designed to do only with homogeneous texture segments, it is not applicable for extraction of such inhomogeneous (coarse) texture area. Therefore, the segmentation procedure should be terminated. And we can conclude that all noticeable homogeneous texture segments have already been extracted in the image.

Algorithm 1: Finding the representative pixel of the most homogeneous image segment.

```
Input:   array Q[Z][I][J],   // Binary representation of texture windows containing in each
                             // z-th test window.
         array F[I][J][M],   // Feature description of the whole image.
         array S[Z][I][J],   // Counters of similar feature patterns.
         H,                  // Predefined threshold.
         M;                  // Number of all texture features (feature types).
Output:  S_max,              // Maximal value of the whole array S.
         X_Coord,            // X coordinate of representative pixel.
         Y_Coord;            // Y coordinate of representative pixel.

S = 0;
for (z = 0;z < Z; z++)                          // Cycle through all test windows.
{
    for (i = 0; i < I; i++)                     // Cycle through X coordinates of whole image.
        for (j = 0; j < J; j++)                 // Cycle through Y coordinates of whole image.
        {
            if (Q[z][i][j] == 1)  // Testing whether this texture window is inside z-th test window.
            {
                for (x = 0; x < I; x++)         // Cycle through X coordinates of whole image.
                    for (y = 0; y < J; y++)     // Cycle through Y coordinates of whole image.
                    {
                        if (Q[z][x][y] == 1)
                        {
                            SUM = 0;
                            for (m = 0; m < M; m++)     // Cycle through all texture features.
                                SUM = SUM + 1 (H – | F[i][j][m] – F[x][y][m] |);
                            S[z][i][j] = S[z][i][j] + 1 (SUM – M + 1);
                        }
                    }
            }
        }
}

// Finding maximal value of array S[Z][I][J] and coordinates of representative pixel.
S_max = 0;
for (z = 0;z < Z; z++)                          // Cycle through all test windows.
    for (i = 0; i < I; i++)                     // Cycle through X coordinates of whole image.
        for (j = 0; j < J; j++)                 // Cycle through Y coordinates of whole image.
        {
            if (S[z][i][j] > S_max)
            {
                S_max = S[z][i][j];
                X_Coord = i;
                Y_Coord = j;
            }
        }
```

5 Experiments

For experiments, a computer program has been created which implements the procedure of finding such a feature pattern that has the maximal number of similar feature patterns among all test windows covering the image. Examples of application of this program to a grayscale natural images of 427 × 320 pixels are shown in Figs. 1 – 4. Every figure consists of upper and lower parts. The entire input image is placed in the upper part of each figure. In the lower part, a single simply connected texture segment

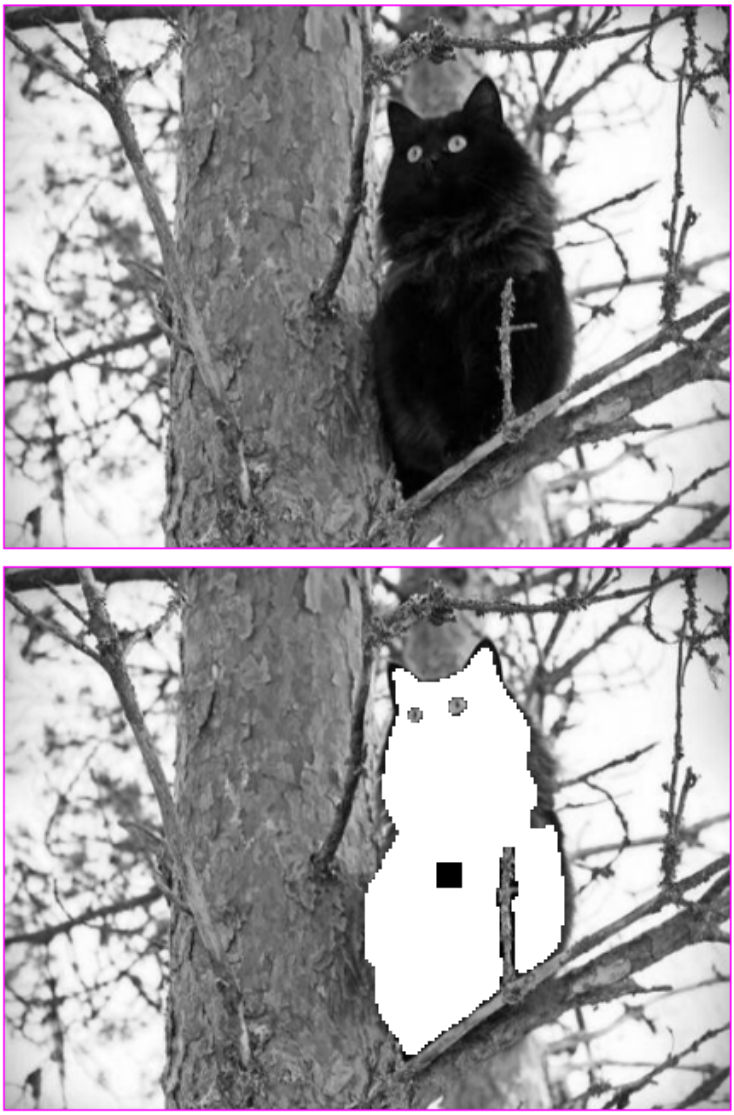


Fig. 1. The upper part is a photograph of a black cat on a tree. In the lower part, location of the representative pixel found by the algorithm is indicated by a black square. This pixel is a representative of the most homogeneous texture segment present in the image. The segment which the algorithm considers as most homogeneous is depicted by overlaying white on the input image in the lower part of the figure. This segment is delineated by the segmentation model using the feature patterns extracted from the found representative pixel and its nearest neighborhood. The parameter $H = 2$.

Fig. 2. The upper part is a photograph of a black cat in a tree. In the lower part, location of the representative pixel found by the algorithm is indicated by a black square. This pixel is a representative of the most homogeneous texture segment present in the image. The segment which the algorithm considers as most homogeneous is depicted by overlaying white on the input image in the lower part of the figure. This segment is delineated by the segmentation model using the feature patterns extracted from the found representative pixel and its nearest neighborhood. The parameter $H = 8$.

Fig. 3. The upper part is a landscape photograph. In the lower part, location of the representative pixel found by the algorithm is indicated by a black square. This pixel is a representative of the most homogeneous texture segment present in the image. The segment which the algorithm considers as most homogeneous is depicted by overlaying white on the input image in the lower part of the figure. This segment is delineated by the segmentation model using the feature patterns extracted from the found representative pixel and its nearest neighborhood.

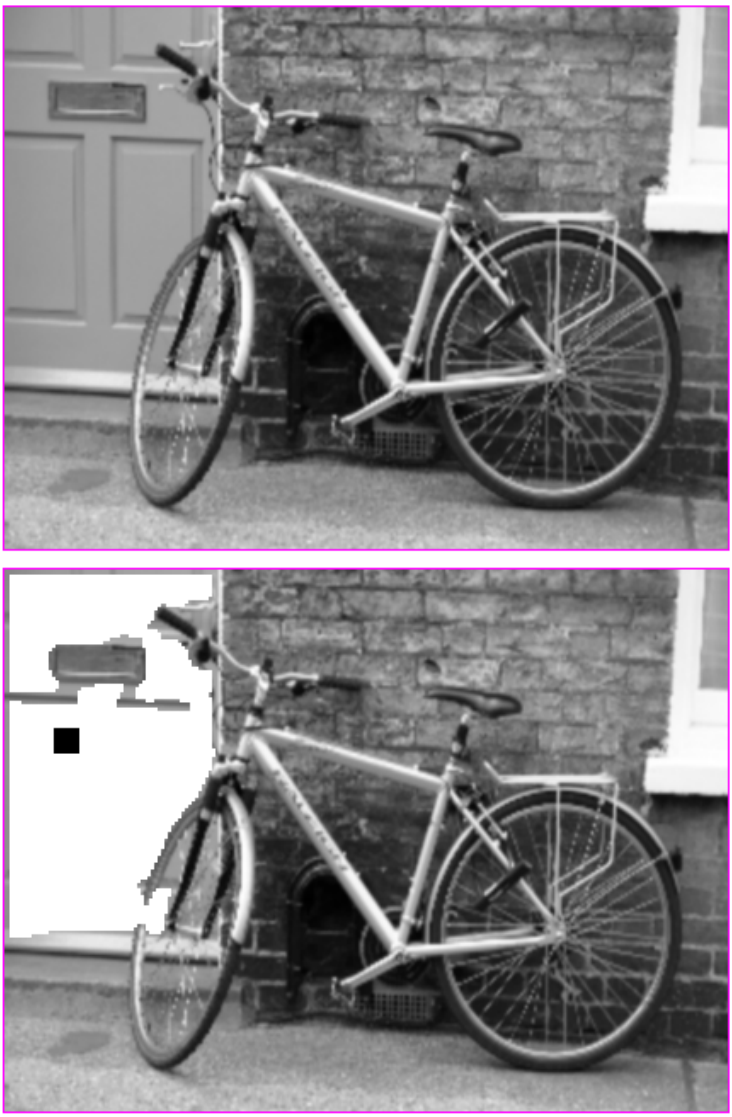


Fig. 4. The upper part is a photograph of a bicycle which is leaned against a wall of a house. In the lower part, location of the representative pixel found by the algorithm is indicated by a black square. This pixel is a representative of the most homogeneous texture segment present in the image. The segment which the algorithm considers as most homogeneous is depicted by overlaying white on the input image in the lower part of the figure. This segment is delineated by the segmentation model using the feature patterns extracted from the found representative pixel and its nearest neighborhood.

is presented by overlaying white on the same input image. This segment demonstrates the area which the algorithm considers as most homogeneous in the image. Location of the representative pixel found by the computer program in the input image is indicated by a small black square within the white segment. The segment is delineated by the segmentation model using the feature patterns extracted from the found representative pixel and its nearest neighborhood. Texture window of 11×11 pixels is applied.

According to above description, the parameter H determines a degree of texture homogeneity of the segment which should be extracted by the segmentation model basing on the feature pattern of the representative pixel. The smaller the value H, the more homogeneous segment should be extracted. The larger the value H, the more coarse texture segment should be extracted. This is illustrated by Figs. 1 and 2, in which the same image is processed with different values of H. In Fig. 1, $H = 2$, and in Fig. 2, $H = 8$. As seen in these figures, much more homogeneous segment of a black cat is extracted in Fig. 1 versus a higher variable texture segment of a tree trunk presented in Fig. 2.

Changing the test window size affects the results of the image processing by the program so that a smaller size leads to extraction of smaller homogeneous texture segments present in the image and vice versa.

6 Discussion and Conclusions

In this work we propose the algorithm for finding the texture feature pattern that characterizes the most homogeneous texture area present in the analyzed image. A series of experiments has been performed that demonstrate efficiency of this algorithm. Four examples of such experiments are presented in the paper.

An important advantage of the algorithm is that it processes all input images without any preliminary learning. Also, the algorithm does not need any prior knowledge about the input image. Instead, the algorithm requires only a desired depth of texture analysis to be specified, which is the similarity threshold of texture features (parameter H, Section 4). In general, this parameter determines a degree of homogeneity of the texture segment (more or less homogeneous or more or less coarse) which the algorithm chooses for extraction. Influence of H on choosing the most homogeneous segment in the image is demonstrated by Figs. 1 and 2.

Another parameter that strongly influences choice of the most homogeneous segment and location of its representative pixel is the texture window size in relation to the image size. Generally, the use of texture window of less size leads to choosing more homogeneous segment of smaller size. However, too small window does not provide an adequate description of textures. Larger window size gives rise to choosing less homogeneous and coarser texture segment.

Selection of all model parameters is quite experimental. In general, its purpose is to adjust the segmentation model so that it could delineate homogeneous texture areas in the same manner as humans. This means that in many experiments we vary and test different values of the parameters with the aim to obtain the best coincidence between the homogeneous segment extracted by the model and the same segment extracted by

a human. In series of experiments with the computer program implementing the algorithm, it demonstrates a reasonable choice of the representative pixel characterizing the most homogeneous texture segment present (among others) in the input image. This is always the case when the image does contain rather sharp-cut homogeneous texture segments. The higher the degree of homogeneity of a texture area, the higher the probability of correct location of its representative pixel by the algorithm.

Nevertheless, in some cases, the segmentation model does not extract correctly such areas that humans consider as homogeneous. This fact means that the proposed feature set does not describe textures completely in the same way as human. However, humans use their understanding of homogeneity which always includes some high-level features, such as the context of the analyzed scene and the knowledge of belonging to a certain object previously recognized in the image. This may explain the differences between the homogeneous segments extracted by the model and those that are correct from a human point of view.

The proposed algorithm is not too computationally expensive The runtime depends on the desired accuracy of texture analysis, videlicet, the total number of test windows Z and its size D. For example, the (unoptimized for speed) code runs about 10 sec on 2GHz PC to extract all texture features and to find the representative pixel of Fig. It is also worth to note that the methods proposed in [31–33] could significantly accelerate the computation of the algorithm as well as the whole segmentation model.

Thus, the proposed algorithm solves some intermediate task within the complex image segmentation problem. The algorithm may be useful for any segmentation technique because it considerably facilitates solution of the whole segmentation problem by providing helpful additional information about the analyzed image. In particular, at present, the algorithm is used as a part of a texture segmentation model which performs sequential extraction of all homogeneous fine-grained texture segments in any input image. The model will be described in details in subsequent papers.

This work was supported in part by the projects UNAM-DGAPA-PAPIIT IN 102014 and UNAM-DGAPA-PAPIIT IT 102814.

References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 888–905 (2000)
2. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, 603–619 (2002)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision **59**, 167–181 (2004)
4. Gao, C., Zhow, D., Guo, Y.: Automatic iterative algorithm for image segmentation using a modified pulse-coupled neural network. Neurocomputing **119**, 332–338 (2013)
5. Bhosle, V.V., Pawar, V.P.: Texture segmentation: different methods. International Journal of Soft Computing and Engineering (IJSCE) **3**, 69–74 (2013)
6. Khan, M.W.: A survey: Image segmentation techniques. International Journal of Future Computer and Communication **3**, 89–93 (2014)
7. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. International Journal of Computer Vision (IJCV) **43**, 7–27 (2001)

8. Wolf, L., Huang, X., Martin, I., Metaxas, D.: Patch-Based Texture Edges and Segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 481–493. Springer, Heidelberg (2006)
9. Caenen, G., Ferrari, V., Zalesny, A., Van Gool, L.: Analyzing the layout of composite textures. In: 2002 International Workshop on Texture Analysis and Synthesis, pp. 15–20 (2002)
10. Alpert, S., Galun, M., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: 2003 IEEE International Conference on Computer Vision (ICCV), pp. 716–723 (2003)
11. Donoser, M., Bischof, H.: Using covariance matrices for unsupervised texture segmentation. In: 2008 International Conference on Pattern Recognition (ICPR), pp. 1–4 (2008)
12. Todorovic, S., Ahuja, N.: Texel-based texture segmentation. In: 2009 IEEE International Conference on Computer Vision (ICCV), pp. 841–848 (2009)
13. Tivive, F.H.C., Bouzerdoum, A.: Texture classification using convolutional neural networks. In: 2006 IEEE Region 10 Conference, pp. 1–4 (2006)
14. Melendez, J., Puig, D., Garcia, M.A.: Multi-level pixel-based texture classification through efficient prototype selection via normalized cut. Pattern Recognition **43**, 4113–4123 (2010)
15. Al-Kadi, O.S.: Supervised texture segmentation: a comparative study. In: 2011 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1–5 (2011)
16. Kussul, E.M., Rachkovskij, D.A., Baidyk, T.N.: On image texture recognition by associative-projective neurocomputer. In: Intelligent Engineering Systems through Artificial Neural Networks Conference (ANNIE), pp. 453–458 (1991)
17. Kussul, E.M., Baidyk, T.N., Lukovich, V.V., Rachkovskij, D.A.: Adaptive neural network classifier with multifloat input coding. In: 6-th Intern. Conf. on Neural Networks and their Industrial and Cognitive Applications (Neuro-Nimes 1993), pp. 25–29 (1993)
18. Goltsev, A.: An assembly neural network for texture segmentation. Neural Networks. **9**, 643–653 (1996)
19. Lukovich, V.V., Goltsev, A.D., Rachkovskij, D.A.: Neural network classifiers for micromechanical equipment diagnostics and micromechanical product quality inspection. In: 5-th European Congress on Intelligent Techniques and Soft Computing (EUFIT 1997), vol. 1, pp. 534–536 (1997)
20. Kussul, E.M., Kasatkina, L.M., Rachkovskij, D.A., Wunsch, D.C.: Application of random threshold neural networks for diagnostics of micro machine tool condition. In: IJCNN 1998, vol. 1, pp. 241–244 (1998)
21. Goltsev, A.D.: Neural Networks with the Assembly Organization, Naukova Dumka, Kiev, Ukraine, p.. 200 (2005). (in Russian)
22. Baidyk, T., Kussul, E., Makeyev, O.: Texture recognition with random subspace neural classifier. In: WSEAS International Conference on Systems Science and Engineering, pp. 319–325 (2005)
23. Makeyev, O., Sazonov, E., Baidyk, T., Martin, A.: Limited receptive area neural classifier for texture recognition of mechanically treated metal surfaces. Neurocomputing **71**, 1413–1421 (2008)
24. Kussul, E.M., Baidyk, T.N., Wunsch, D.C.: Neural Networks and Micro Mechanics, p. 210. Springer (2010). ISBN 978-3-642-02534-1
25. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based feature space. In: 2003 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 699–704 (2003)

26. Clausi, D.A., Deng, H.: Design-based texture feature fusion using Gabor filters and co-occurrence probabilities. IEEE Transactions on Image Processing **14**, 925–936 (2005)
27. Wei, H., Bartels, M.: Unsupervised segmentation using Gabor wavelets and statistical features in LIDAR data analysis. In: 2006 International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 667–670 (2006)
28. Yang, A.Y., Wright, J., Ma, Y., Shakar, S.: Sastry, Unsupervised segmentation of natural images via lossy data compression. Computer Vision and Image Understanding **110**, 212–225 (2008)
29. Comaniciu, D.: An algorithm for data-driven bandwidth selection. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**, 1–8 (2003)
30. Mahbubur Rahman, M.: Unsupervised natural image segmentation using mean histogram features. Journal of Multimedia **7**, 332–340 (2012)
31. Rachkovskij, D.A., Misuno, I.S., Slipchenko, S.V.: Vector data transformation using random binary matrices. Cybernetics and Systems Analysis **48**, 146–156 (2012)
32. Rachkovskij, D.A., Kussul, E.M., Baidyk, T.N.: Building a world model with structure-sensitive sparse binary distributed representations. Biologically Inspired Cognitive Architectures **3**, 64–86 (2013)
33. Gritsenko, V.I., Rachkovskij, D.A., Goltsev, A.D., Lukovych, V.V., Misuno, I.S., Revunova, E.G., Slipchenko, S.V., Sokolov, A.M., Talayev, S.A.: Neural distributed representation for intelligent information technologies and modeling of thinking. Cybernetics and Computer Engineering **173**, 7–24 (2013). (in Russian)

Robust Tracking for Augmented Reality

José M. González-Linares(✉), Nicolás Guil, and Julián Ramos Cózar

Department of Computer Architecture, University of Málaga, Málaga, Spain
jgl@uma.es

Abstract. In this paper a method for improving a tracking algorithm in an augmented reality application is presented. This method addresses several issues to this particular application, like marker-less tracking and color constancy with low quality cameras, or precise tracking with real-time constraints. Due to size restrictions some of the objects are tracked using color information. To improve the quality of the detection, a color selection scheme is proposed to increase color distance between different objects in the scene. Moreover, a new color constancy method based in a diagonal-offset model and k-means is presented. Finally, some real images are used to show the improvement with this new method.

1 Introduction

Augmented reality (AR) applications are gaining popularity due to their huge potential impact on many areas, including medicine, manufacturing, robotics, entertainment, tourism or education. Most AR applications need some enabling technologies including displays, tracking, registration and calibration (Azuma, y otros, 2001). In this work we focus on improving tracking in an AR education application, but our proposal can be easily adapted to other AR applications.

AR education applications allow educators to create a scenario, provide specific information for that scenario, and even embed data seamlessly within the real world context (Bower, Howe, McCredie, Robinson, & Grover, 2013). In this context Seabery, a Spanish technological company, has developed a tool for augmented training for welding called Soldamatic. This application uses a head display and several real objects like welding torches or work pieces to recreate a realistic welding training experience.

One of the most difficult stages in this application is robust and precise tracking of a long rod that simulates the electrode used in the welding process. This rod has to be located as precisely as possible to obtain a good estimation of the welding procedure. Unfortunately, its elongated and thin shape makes very difficult the use of markers that are used in other type of pieces (e.g. probes). Moreover, edges obtained in this small rod are not very precise because the spatial resolution of the camera. Thus, a color based approach has been devised.

This application is used in many different real environments, and the light and working set can vary from one place to another. Differences in illumination cause deviations of the colors measurements that can be solved using a color constancy algorithm (Buchsbaum, 1980). Most color constancy algorithms are based in some assumptions: usually only a single spatially uniform illuminant is considered, and objects are assumed to be flat, coplanar, and Lambertian. This last assumption doesn´t hold in this application because objects are very close to the camera and their angle of view can vary a lot.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 301–308, 2015.
DOI: 10.1007/978-3-319-19258-1_26

We have tested several color constancy algorithms with poor results, thus we have developed a new method that obtains better results for this specific problem.

The rest of the paper is organized as follows. A method for selecting colors of the objects in the scene is presented in Section 2. Then, a new algorithm for color constancy based in the diagonal-offset method and k-means is described in Section 3. In Section 4, a method for the detection and location of an elongated object is shown. Finally, in Section 5 some conclusions and future work are presented.

2 Colors Selection

Color selection is a crucial issue as the quality of rod color segmentation procedure highly relies on a good distinction among the rod colors and the others elements that typically appear in the scene. These are mainly the probes, colored in blue (Pantone 3015) and green (Pantone 363) as it is shown in Figure 2. Other relevant colors are the black and white of the markers and the red (Pantone 484) of the point of the welding gun.

After several experiments we decided four different colors would be used for rod point localization and orientation computation. Minimizing the number of different colors used in the rod allow us to choose a more contrasted palette that reduces errors during the segmentation step. In addition, different lengths of the color markers are used in order to facilitate the location of the rod point.

Therefore, color selection involves solving a placement problem: given the most frequently used colors, estimate the coordinates of new colors that maximizes the distance among the resulting color set (given and new). For practical reasons the xyz color coordinate system has been chosen, because in this space a good approximation for measuring color differences is obtained using the Euclidean distance.

The solution must be subject to an important restriction: not every reproducible color in the xyz system can also be reproduced in a printing system. The paint industry standard that offers one of the richer chromatic varieties is a color palette called Pantone. This color card offers a range of about 1,000 different colors. Thus our solution must be one color in this set. Figure 1 (left) represents the whole color set in the xy plane. It is remarkable that the entire gamut is bounded by a triangle. The figure also includes the position of the five given colors.

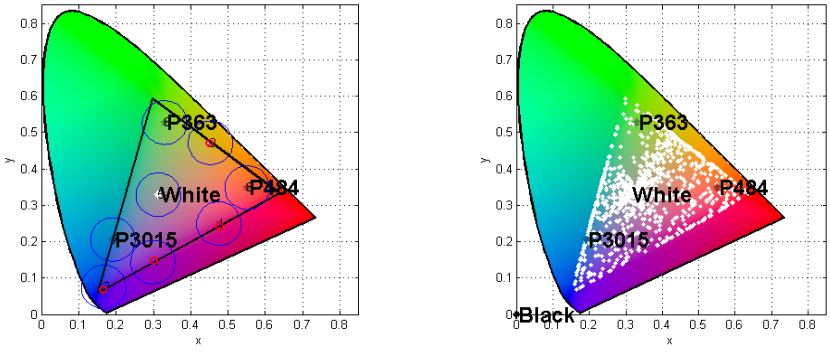


Fig. 1. Pantone full color set and position of the most frequent colors in the scene. Selected colors.

The problem can be stated more precisely as follows. Find the position of the center of a set of four circles that minimizes the area of the intersection among these circles and the circles defined by the position of the five fixed colors, subject to the center locations must establish a valid Pantone color.

The expression that relates relative position of the circles centers with the area enclosed by their intersection is a highly non-linear formula. Therefore, a genetic optimization algorithm has been selected to efficiently find the optimal placement. An important parameter is the diameter of the circles. Its length represents the distance from which we consider the colors are clearly distinguishable. In one hand, a low value can produce a solution with very similar colors, but on the other hand, a high value can turn the problem to unfeasible. Thus the optimization has been tested with a range of values for the radius R and we have discovered that a good compromise is obtained with R=0.06.

The center location restriction is imposed in two steps. In a first stage, circles centers are forced to lie inside the triangle that bounds the Pantone color set. In the second step, the nearest neighbor algorithm is used to find the Pantone color set nearest to the solution. If the distances were not small enough, the optimization algorithm with different parameters would be run again. However, as it is expected, solutions tend to be at the triangle borders, where the color availability is maximum. We have preferred to impose the restriction on the points lying inside the triangle rather than to allow only coordinates of Pantone colors during the optimization process as the first option can be more easily adapted to add new restrictions to the color set based on their visual appearance. The nonlinear constrains to restrict the solution to the points P inside a triangle defined by points A_1, A_2 and A_3, if it is based on sub-triangle orientations, can be written as follows. Let O_{123}, O_{12P}, O_{23P} and O_{31P} be the determinant of the matrix formed by stacking the coordinate differences for different pairs of points. More precisely

$$O_{123} = det \begin{vmatrix} A_1 - A_3 \\ A_2 - A_3 \end{vmatrix},$$

$$O_{12P} = det \begin{vmatrix} A_1 - P \\ A_2 - P \end{vmatrix}, O_{23P} = det \begin{vmatrix} A_2 - P \\ A_3 - P \end{vmatrix} \text{ and } O_{31P} = det \begin{vmatrix} A_3 - P \\ A_1 - P \end{vmatrix}.$$

where P is inside the triangle defined by $\widehat{A_1A_2A_3}$ if $sign(O_{123}) = sign(O_{12P}) = sign(O_{23P}) = sign(O_{31P})$, i.e., the sign of these determinants are the same.

Figure 1 (right) depicts the solution found by the genetic algorithm after running for a few hundred generations. The red symbol ‘o’ indicates the precise location of a valid Pantone color.

3 Color Constancy

A video shot captured by the camera is shown in Figure 2. Several pixels have been annotated to show the deviation of the blue (Pantone 3015) and green (Pantone 363)

colors under real illumination conditions. Blue pixels are relatively close to their ideal value, but green pixels have been biased and now they are closer to the blue values. Depending on the viewing angle and non-uniformity of the illuminant, this bias can be more or less high.

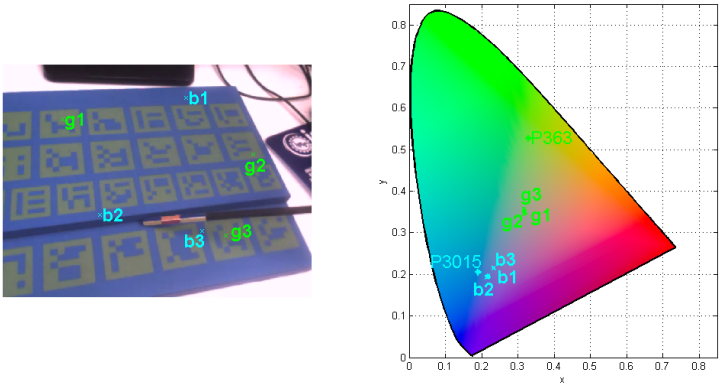


Fig. 2. Video shot captured by the system. Chromaticity values of several green (g1, g2 and g3) and blue (b1, b2 and b3) points are shown in a CIE chromaticity diagram in the right**.**

Most color constancy algorithms use the diagonal transform or von Kries Model (Hordley, 2006) to map the image gamut into a canonical gamut when certain conditions are me. More precisely, if only a single spatially uniform illuminant is considered, and objects are flat, coplanar, and Lambertian, that is, their reflectances are diffuse and independent from the angle of view. Unfortunately these conditions are not fulfilled in our experiments and, consequently, the diagonal transform cannot successfully map the image gamut into the canonical gamut. Thus, we have selected another model, the diagonal-offset model (Finlayson, Hordley, & Xu, 2005), which considers six parameters to map the image gamut. Let $[R^c\ G^c B^c]$ be a color of the canonical gamut and $[R^o\ G^o B^o]$ an observed color of the image gamut, then the diagonal-offset model maps this color using the next equation:

$$\begin{bmatrix} R^c \\ G^c \\ B^c \end{bmatrix} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} R^o \\ G^o \\ B^o \end{bmatrix} + \begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} \tag{1}$$

These six parameters can be obtained using convex programming but we have devised another method to reduce computational requirements and to improve the solution.

From our setup we know that most of the image is occupied by the blue and green working piece, plus the welding torch whose colors were selected in Section 2. Thus, the gamut of a typical frame in our application contains just a few colors, and some of them are known beforehand. Then, if we cluster the image gamut in several groups using k-means, for example with k=10, these clusters will likely correspond to the known colors and, moreover, two of the largest clusters will correspond to P3015 and P363 because the working piece occupies most of the frame.

We can select the centroids of the largest groups and substitute them into Equation 1, altogether with the RGB values of the green and blue Pantone colors, to get several six equations systems that allow directly computing the six parameters. For example, given the RGB values of P363 [76,140,43], P3015 [0,98,155], and the centroids of two clusters $i\ [R^i, G^i, B^i]$ and $j\ [R^j, G^j, B^j]$, we can easily compute $[d_1, d_2, d_3]$ and $[o_1, o_2, o_3]$ by solving the next system:

$$76 = d_1 * R^i + o_1$$
$$140 = d_2 * G^i + o_2$$
$$43 = d_3 * B^i + o_3$$
$$0 = d_1 * R^j + o_1$$
$$98 = d_2 * G^j + o_2$$
$$155 = d_3 * B^j + o_3$$

Not every solution found by this way is a feasible illuminant, for example solutions with negative scaling factors ($d < 0$) are unreal, thus some of them can be automatically discarded. On the other hand, some of the remaining solutions correspond to wrong illuminants, for example illuminants that transform real red values into blue values, and so on. These solutions can be discarded by computing the distance between the observed values and their expected canonical values, and removing those that are farther. And finally, as it is recommended in (Finlayson, Hordley, & Xu, 2005), we should look up among the remaining feasible solutions for the solution with a lowest offset vector.

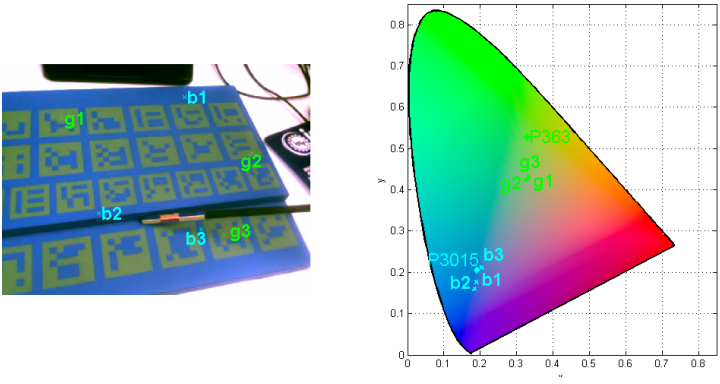


Fig. 3. Result of color correction algorithm on image of Fig. 1. Blue points are slightly closer to Pantone 3015, and green points are much closer to Pantone 363.

In Figure 3 it is shown the results obtained with this method in the image of Figure 2. Blue points remain close to Pantone 3015 and green points have been corrected to a much closer position to Pantone 363.

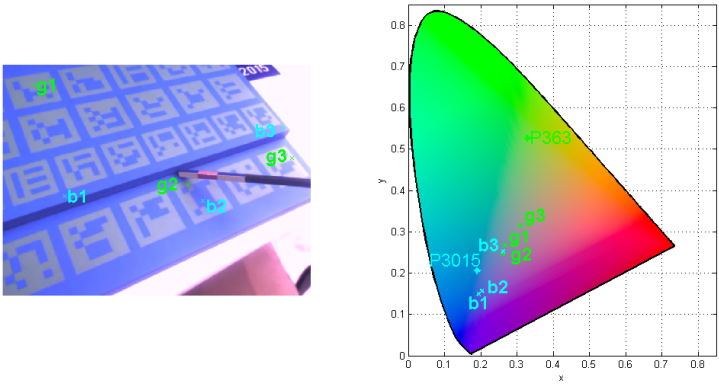


Fig. 4. A video shot under a bluish illuminant. Green and blue points are closer to Pantone 3015, and even one of the green points (g2) is closer than one of the blue points (b3).

A worse situation is shown in Figure 4. The scene was shot using a non-uniform bright bluish illuminant and most colors had been biased to the blue area. There are green points like *g2* that are closer to Pantone 3015 than other blue points like *b3*, rendering markers detection more difficult. Nevertheless, our color constancy method obtains a good result as it is shown in Figure 5. Green points in the left part of the image are now very close to Pantone 363, and the rest of green points are closer than before.

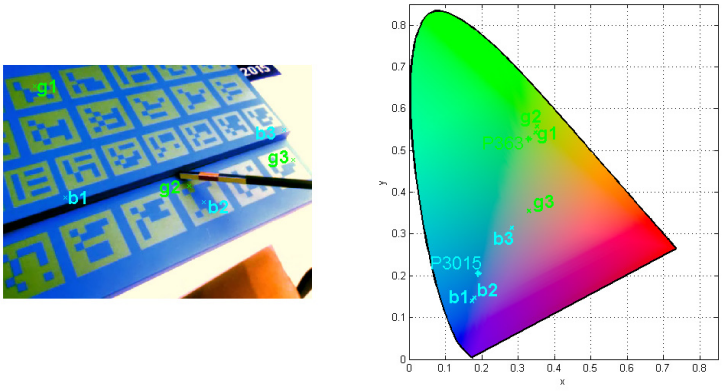


Fig. 5. After correcting image in Fig. 3, the green points are closer to P363 and the blue points have not been affected much.

Finally, to assess the computational requirements of our method, we have implemented it using the OpenCV library to measure the execution time. A full augmented reality application is typically multi-threaded, with specific threads for frame acquisition, processing, visualization, etc., thus we have developed a single-threaded version

that can be easily incorporated into the full application. We have measured execution times between 5 and 7 ms in an Intel Xeon X5690 using RGB color frames of size 640x480, therefore real-time constraints can be met with our algorithm.

4 Rod Segments Detection

Once the image has been corrected, the rod segments can be more easily detected. A different color was assigned to each segment as it was described in Section 2. To select which pixels belong to a specific color we use the distance metric defined by CIE (International Color Consortium), the CIEDE2000 color-difference formula (Sharma, Wu, & Dalal, 2005). For each color a mask is computed where pixels whose distance is below a threshold are selected. These pixels are grouped in blobs that correspond to the rod segments. They are somehow irregular and do not exactly represent a rectangular area as expected because, on the one hand, their size is too small for the spatial resolution of the camera and, on the other hand, the edges of the rod are not perfectly straight. Nevertheless, we can fit these blobs to rectangular areas and adjust the gaps by considering some geometrical constraints.

First, we fit each blob to an ellipse using the method presented in (Fitzgibbon, Pilu, & Fisher, 1999). The ellipsis of blobs corresponding to rod segments should be approximately aligned and very close among them. If there is a small gap between two contiguous ellipsis, they are both elongated to obtain two touching ellipsis. Then, the rotated rectangles that contain these ellipses can be computed to obtain the rod segments.

Fig. 6. Result obtained searching for the rod segments. Left, color corrected video short. Center, ellipsis detected. Right, corners of rod segments.

In Figure 6 it is shown the result of searching for Pantone 395 (lime) and Pantone 484 (red) in a video frame. Color corrected video shot is shown in the left part of the figure. In the middle part, the ellipsis corresponding to the three rod segments are drawn. These ellipsis are not perfectly aligned and do not entirely occupy the segment area, thus they are elongated to obtain a better approximation. In the right part of Figure 6, the corners of the detected rod segments are marked.

Although there are some small errors in the location of the corners, they can be cancelled when the homography is computed for all these points.

5 Conclusions

In this paper we have presented a method for improving a tracking algorithm in an augmented reality application. Our method addresses several issues to this particular application, like marker-less tracking and color constancy with low quality cameras, or precise tracking with real-time constraints. We have presented a method for selecting colors of the objects in the scene and a new algorithm for color constancy based on the diagonal-offset method and k-means. Our method significantly improves the color information in the scene, making feasible a color based tracking algorithm.

In a future work we pretend to improve our color constancy method using other illumination models. We also plan to enhance tracking precision by using super-resolution techniques.

Acknowledgements. This work has been partially supported by the Ministry of Education of Spain (TIN2013-42253P) and the Junta de Andalucía of Spain (TIC-1692). We also gratefully thank Seabery for providing the camera and equipment.

References

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. Computer Graphics and Applications **21**(6), 34–47 (2001)

Bower, M., Howe, C., McCredie, N., Robinson, A., Grover, D.: Augmented reality in Education - Cases, places, and potentials. Educational Media (ICEM), pp. 1–11. IEEE, Singapore (2013)

Buchsbaum, G.: A spatial processor model for object colour perception. Journal of the Franklin Institute, 1–26 (1980)

Finlayson, G., Hordley, S., Xu, R.: Convex Programming Colour Constancy with a Diagonal-Offset Model. International Conference on Image Processing, pp. 948–951 (2005)

Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct Least Square Fitting of Ellipses. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(5), 476–480 (1999)

Hordley, S.: Scene Illumination Estimation: Past, Present and Future. Color Research & Application **31**, 303–314 (2006)

Sharma, G., Wu, W., Dalal, E.: The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research & Application **30**, 21–30 (2005)

Bio-inspired Motion Estimation with Event-Driven Sensors

Francisco Barranco[1,2(✉)], Cornelia Fermuller[1], and Yiannis Aloimonos[1]

[1] University of Maryland, UMIACS, CFAR, A.V. Williams Bldg, College Park, MD, USA
{fbarranco,cfer,yiannis}@umiacs.umd.edu
[2] University of Granada, CITIC, ETSIIT, Daniel Saucedo Aranda sn, Granada, Spain

Abstract. This paper presents a method for image motion estimation for event-based sensors. Accurate and fast image flow estimation still challenges Computer Vision. A new paradigm based on asynchronous event-based data provides an interesting alternative and has shown to provide good estimation at high contrast contours by estimating motion based on very accurate timing. However, these techniques still fail in regions of high-frequency texture. This work presents a simple method for locating those regions, and a novel phase-based method for event sensors that estimates more accurately these regions. Finally, we evaluate and compare our results with other state-of-the-art techniques.

Keywords: Bio-inspired systems · Neuromorphic engineering · Motion estimation · Event-driven sensors · Asynchronous sensors

1 Introduction

Biological strategies refined by evolution, have been emulated in various fields and been applied to a wide range of application [18]. We are interested in the adaption of biological vision neural systems to computational principles for the estimation of visual features. Current cameras are operating synchronously, acquiring image frames at a fixed sampling frequency, but capture visual information in a continuous world that works asynchronously. This quantization limits applications requiring high-speed maneuvering in autonomous navigation and robotics. When there are no changes, the same visual information is recorded in different frames. This translates into a demand for plenty resources and memory in order to process redundant data.

In the last few years frame-free sensors have become increasingly popular due their accurate time triggering, low-latency, real-time performance, and low-resource requirements. Emulating the human neural vision system, these sensors are driven by events. There are two main retina-brain pathways in the human early vision system: the sustained pathway, that is believed to conduct information such as color, texture, or intensity patterns, and the transient pathway, that

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 309–321, 2015.
DOI: 10.1007/978-3-319-19258-1_27

only responds to changes. Sensors have focused on the first one while frame-free sensors emulate the second. In our case, the DVS (Dynamic Vision Sensor [15]) triggers new events when the reflectance for a specific location changes. In case of no change, no information is generated. Instead of an external clock that defines the sampling frequency, the control is transferred to the individual pixels that handles it independently.

From the Computer Vision point of view, the efforts in motion estimation have been devoted to improving accuracy. The main challenges are due to the difficulty in estimating at object boundaries, where there are discontinuities in the flow field. Here is where event-based computation represents an alternative, since motion estimation is very accurate at boundaries even when the contours of the objects are not provided in advance. This is because of the high working frequency of these sensors that allow to obtain image motion by observing events between neighboring pixels. Thus one can obtain image motion only from edges without crossing different moving regions.

Current motion estimation techniques for event sensors provide accurate results, even allowing us to track pixels between neighboring positions (especially in [3,6]). Their underlying assumption is that events that are fired during a short period of time at close-by positions are due to changes in the same edge. Consequently, these methods work accurately for strong contrast edges. However, high-frequency textures violate the assumption. At such locations, frequency-based approaches can increase the accuracy. This paper presents an event-driven method based on local phase (in the frequency domain). Moreover, we also describe a method to localize the contours of textured objects. Our evaluation shows that the method achieves more accurate estimates for highly textured areas than previous approaches.

The paper is structured as follows: section §2 describes the sensor and Section §3 motivates the motion estimation. Section §4 describes the contour localization and Section §5 the new phase-based technique for frame-free sensors. Finally, Section §6 presents results and Section §7 gives the conclusions.

2 Asynchronous Frame-Free Sensors

For a conventional camera, a sequence of $M \times N$ frames, are taken at a given sampling rate f_s, such that the value of each pixel (x, y) with $x \in [0, M-1]$ and $y \in [0, N-1]$ accounts for the intensity recorded during a time $\Delta t_s = f_s^{-1}$. The Dynamic Visual Sensor (DVS [15]) fires asynchronous address-events that signal reflectance changes at the time they occur. Its spatial resolution is 128×128 and its maximum temporal resolution is 15 μs. The DVS does not use a fixed sampling rate. Instead, its samples the input every time the variation with respect to the last measured value at the same position (x, y) exceeds a fixed amount. In other words, for an image point (x, y), at time t an event $ev(x, y, t, p)$ is fired, where p is the polarity $+1$ or -1, if the log of the intensity I increases or decreases by a global threshold ΔI_s (see Eq. 1). For example, if as in our case, the threshold is 0.1, it means that the intensity has to change by 10% to fire

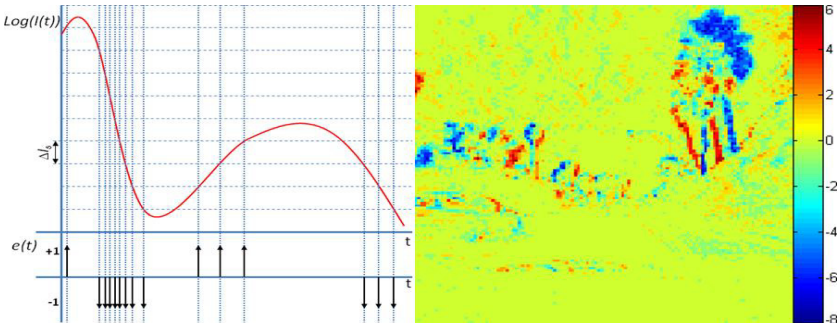


Fig. 1. Left: Top part shows the log Intensity and bottom the events fired when the input changes by ΔI_s. Events are + or - according to the sign of the change. Right: Image to visualize the events accumulated during 50 ms: warm colors represents positive counts, cold colors are negative counts and green, the lack of events.

an event. The logarithmic input helps dealing with high dynamic ranges. Fig. 1 illustrates the workings of the sensor.

$$|\Delta(\log(I(x,y,t)))| > \Delta I_s \tag{1}$$

Asynchronous frame-free sensors help us address some of the challenges of motion processing arising in classical Computer Vision. First, since the DVS only transmits data where and when changes happen, it helps saving a lot of resources. Second, since the events are fired at a very high frequency, with a temporal resolution of a few microseconds, there is no blur due to fast motion. Moreover, because the input is logarithmic, the sensor can deal with high dynamic range.

3 Motion Estimation for Asynchronous Frame-Free Sensors

Conventional motion estimation techniques work well in smooth textured areas, but they have problems at object contours [21]. The greatest challenge are fast motions and particularly scenes with both: large and small motions. On the other hand, data from frame-free sensors is spatially highly localized, and because of the high temporal resolution, potentially allows to locate edges without crossing different moving regions leading to very accurate boundary motion. Furthermore, fast motions are not a problem for these sensors. Because of these advantages, frame-based motion processing has great potential for computational motion processing, either alone or combined with classic processing [3].

In event-based space, only normal flow can be estimated directly. Normal flow is the projection of the flow in the direction of the spatial gradient, and it is sufficient for computing 3D motion as shown in [10,11].

Authors in [7] adapt the Lucas-Kanade method [16]. The Optical Flow Constraint is re-written assimilating the number of events accumulated during a time

interval Δt into intensity. Denoting as $e(\mathbf{x}, t)$ the event that happens at position $\mathbf{x}$ and time t, the derivatives of e are approximated as in Eq. 2, where $t_1 < t$.

$$\nabla e(\mathbf{x}, t) \approx \sum_{t-\Delta t}^{t} e(\mathbf{x}, t) - \sum_{t-\Delta t}^{t} e(\mathbf{x} - (1, 1), t), \qquad e_t(\mathbf{x}, t) \approx \frac{\sum_{t1}^{t} e(\mathbf{x}, t)}{t - t_1} \tag{2}$$

In [6], the authors define a function $\mathcal{T}_e(\mathbf{x})$ that assigns only the time of every new event fired at position $\mathbf{x}$. Then, the gradient vector of $\mathcal{T}_e$ provides the inverse of the local velocity vector as in $\nabla \mathcal{T}_e = (1/\mathbf{v})^T$. The velocity is estimated as the inverse of the horizontal and vertical derivative of this plane. Although accurate, at broad edges different close-by pixels can fire events at almost the same time, making the estimation not reliable.

Finally, [3] addresses the issue of broad edges, and presents an alternative that reconstructs the contrast at object edges and tracks them. The method first locates for a certain time interval all the events of the moving edge, the so-called motion boundary. Then, the velocity is estimated as the ratio of all events in the motion boundary over the events at a single point, normalized for time. Accumulating events over multiple time intervals and averaging provides robustness. This work is also the first that combines asynchronous events and synchronous frames with the new DAVIS sensor [8].

However, all these methods have problems with textures. First, at object contours moving on top of textured regions, the contrast between contour and background changes over a small time interval, causing different amount of events. Second, highly-textured areas introduce too much variability for fitting a plane, or for reconstructing the intensity. Frequency-based methods, adapted to asynchronous event space, can help with this problem.

4 How Texture Affects Event-Based Frameworks

As mentioned, current techniques assume that nearby events, which are fired close-by in time, belong to the same structure. However, in highly textured areas or at broader edges next to textured background, the assumption is violated.

First, let us distinguish between edges and contours: *edges* are all the strong discontinuities in the intensity signal, while *contours* are only those edges that correspond to the boundaries of objects (depth discontinuities).

Next, we examine the different cases imposed by movement and texture. Case 1: Objects moving on static non-textured background. If the object is textured, events from contours and texture edges are similar. Event-based methods work well at contours, but they may not work well in a textured region, if it has high-frequency, and the method accumulates events over longer time intervals. If the object is plain, only at the boundary of the object events are fired, where classic event-based methods work well. Case 2: Moving objects on static textured background. Since the background is textured, the patterns of events on the contours over time are changing. Such situations pose a problem for event based methods. Case 3: Both the object and the textured background are moving.

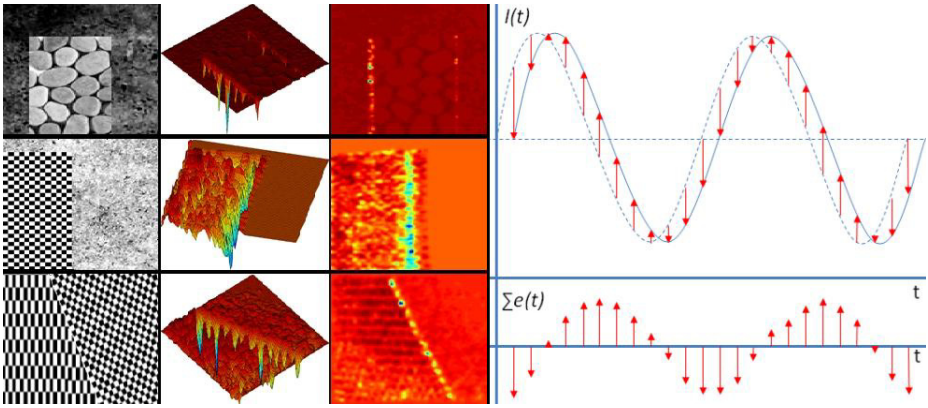


Fig. 2. Left: Contour localization using cross-correlation (left to right): original frame, 3D and 2D representation of the cross-correlation coefficient. Right: A sinusoid moving to the left shown as dashed line at time t and as solid line at $t+\Delta t$. The recorded events over a small time interval make a sinusoid function of same frequency. Red arrows are proportional in size to the amount of positive or negative events at the location.

In this case, two different physical motions take place in regions next to object boundaries. This is the most difficult case for image motion estimation.

4.1 Locating Contours of Textured Objects

Contours can be separated from other edges if we consider the patterns of events at single pixels. The event pattern at texture edges should be shifted (since there is the same contrast between neighboring pixels over time.) But for contours the pattern changes due to a background texture. Considering separately every pixel, the timestamps of the events can be looked at as a 1D signal. Normalized cross-correlation measures the similarity of 1D signals, and it is widely used in signal processing [19]. Our idea is that if there is a texture, events from neighboring positions would have very high cross-correlation coefficients while in the case of broad edges or contours on top of textured backgrounds, the crosscorrelation is low. Let us assume we have the $1 \times N$ signal $f(t)$ and $1 \times M$ signal $g(t)$, Eq. 3 defines the normalized cross-correlation

$$ncc(d) = \frac{|\sum_{t=1}^{t=N}(f(t)-\overline{f})^2(g(t-d)-\overline{g})^2|}{||(f(t)-\overline{f})^2||_2||(g(t-d)-\overline{g})^2||_2} \tag{3}$$

Where $d \in [0, M-1]$ is the lag or delay between the signals (in case they are shifted), and the normalized cross-correlation is an $M \times N$ vector. The maximum gives us the coefficient for measuring the similarity. As mentioned, due to the high-temporal resolution of the sensor, changes are tracked pixel by pixel. Then, the cross-correlation is performed for the 1D signals (composed by the series of timestamps of the events fired) of the neighboring pixels in a region of only

3×3. Next, to increase the robustness, more events are required since noise and distortions may limit the accuracy. For signal correlation, low-pass filters reduce signal noise. However, asynchronous frame-free sensors fire discrete events. Thus, instead of low-pass filters we use activity filters to reduce outliers and noise. Events do not occur isolated in time and/or space, so events that are not very close spatio-temporally to other groups of events are discarded (we use 5 ms intervals and 3×3 windows). Moreover, robustness increases with the number of samples. Assuming longer time intervals increases the latency and reduces performance. In our work, we propose using larger spatial windows (3×3) and compact all their events into a single 1D signal.

The left part of Fig. 2 shows the cross-correlation coefficient. The first row shows a synthetic sequence with a texture moving on top of a textured background (case 1). The first image shows the original frame, the last two images illustrate the cross-correlation coefficient with 3D and 2D views. The minimum correlation values show the object boundaries (in yellow/green). Row 2 shows an example for case 2. There is negative correlation, since for real-world sequences, the background generates a lot of activity at the contour of the object due to the texture and noise. Finally, row 3 shows an example for case 3, in which the negative cross-correlation coefficient values separate two different textures.

5 Phase-Based Motion Estimation

This work presents for the first time the use of frequency-based concepts for motion estimation with event-based sensors. Frequency-based methods or energy-based methods on classical image frames use the responses from a bank of filters tuned to different spatio-temporal frequencies. Different methods consider the energy of these responses [1,13,25]. These techniques usually estimate at a point the local spatial and temporal angular frequencies by computing responses to a bank of filters tuned to different frequencies. In our case, instead of directly using the frequency, we use the local phase [12]. Since the Gabor responses from the filter bank are complex valued, they can be expressed as in Eq. 4

$$g(\mathbf{x},t) = Gabor(\mathbf{x},t) * I(\mathbf{x},t) = \rho(\mathbf{x},t)e^{i\phi(\mathbf{x},t)} \tag{4}$$

where ρ is the amplitude and ϕ the phase. Therefore, the filter responses can be used to extract the local phase $\phi(\mathbf{x},t)$ (the angle of the odd and even parts). Now, an adapted Optical Flow Constraint that assumes the constancy of the spatio-temporal contours using the phase is formulated. Due to the aperture problem, only the velocity in the direction of the spatial phase gradient $\nabla_{\mathbf{x}}\phi$ can be computed. In this case, the normal flow $\mathbf{v}_n$ can be estimated as $\mathbf{v}_n = -\phi_t/||\nabla_{\mathbf{x}}\phi||_2$ with ϕ_t the temporal derivative of the phase.

5.1 Frame-Free Estimation

Let us consider for example, a 1D sinusoid that is moving to the left. The events collected over a small time interval Δt are due to the difference of the sinusoid

and its phase-shifted copy, as shown in the right part of Fig. 2 by the solid and dashed lines respectively. The signal of accumulated events shown in the same axis is another sinusoid of the same frequency. For the sake of clarity, this figure ignores that the input is logarithmic for our DVS.

Let us consider that the first sinusoid equation is $acos(\omega x(t))$, with a denoting the amplitude. Then the accumulated events can be expressed as a sinusoid of with a different amplitude but the same frequency than the original, and shifted.

$$S(x(t),t) = acos(\omega x(t)) - acos(\omega x(t)+\Delta t) = 2asin\left(\frac{\Delta t}{2}\right)cos\left(\omega x(t)+\frac{1}{2}(\Delta t+C)\right) \tag{5}$$

The signal of accumulated events (see Eq. 5) is shifted, but this shift is always the same and only depends on Δt. Therefore, an alternative estimation based on the accumulations of events is possible. Moreover, we assume that the local phase of the signal of accumulated events remains constant over a short time interval. The first step consists in defining the function of accumulated events for a small time interval as $\mathcal{S}\colon \mathbb{N}^2, \mathbb{R} \rightarrow \mathbb{N}$, that assigns to every position $\mathbf{x}$ the number of events that occur during that time interval at that position (Eq. 6).

$$\mathcal{S}(\mathbf{x},t) = k \sum_{t-\Delta t}^{t} e(\mathbf{x},t), \qquad k \in \mathbb{R} \tag{6}$$

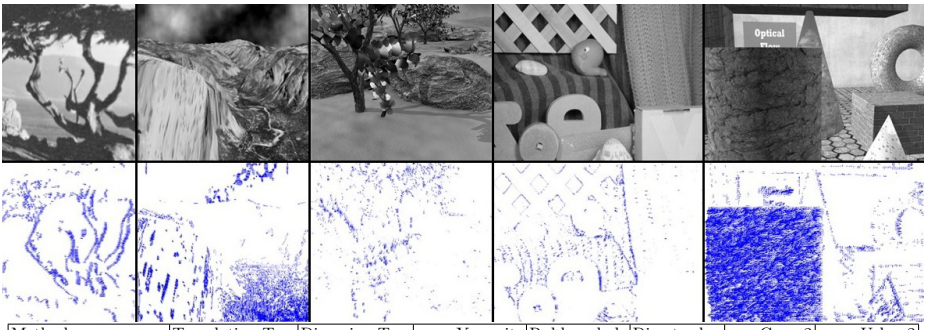
Instead of image intensity we use this signal in the event-based framework, and the assumption of constant local image phase in [12] is rephrased as assuming the constancy of spatio-temporal contours of the signal of accumulated events. This way, the solution for the new Optical Flow Constraint is based on the estimation of the phase gradient of accumulated events and computed as in Eq. 7

$$\nabla\phi(\mathbf{x},t) = \frac{Im[g^*(\mathbf{x},t)\nabla g(\mathbf{x},t)]}{Re[g(\mathbf{x},t)]^2 + Im[g(\mathbf{x},t)]^2}, \tag{7}$$

where $\nabla\phi$ is the phase gradient, with $g(\mathbf{x},t)$ the complex response of the Gabor filter bank for the signal $S(\mathbf{x},t)$, Im and Re are the imaginary and real parts, and g^* stands for the complex conjugate.

6 Results

This section compares our method to other event-based algorithms as well as conventional frame-based methods. We have created three benchmark datasets: in §6.1 using conventional benchmarks and creating synthetic events, in §6.2 using 3D scene model and motion ground-truth and creating synthetic events, and in §6.3 collecting events with the DVS mounted on a robotic mobile platform that provides its egomotion.



Method	Translation Tree	Diverging Tree	Yosemite	Rubberwhale	Dimetrodon	Grove2	Urban2
Event phase	16.6 (9.6%)	**29.8 (5.5%)**	**21.6 (12.4%)**	30.7 (5.2%)	42.5 (1.6%)	23.1 (5.3%)	**21.7 (2.2%)**
Fleet & Jepson [12]	98.4	94.7	97.3	95.4	NA	98.8	99.6
Otte-Nagel [20]	17.6	46.7	39.7	24.9	NA	38.9	86.0
Uras [24]	37.5	47.0	45.1	53.1	54.2	48.7	85.5
Horn-Schuck [14]	47.9	48.7	61.8	34.7	72.0	84.9	94.9
Lucas & Kanade [16]	51.8	55.3	61.3	44.6	61.6	79.2	88.3
Sun [22]	**1.6**	46.6	27.0	**11.6**	**4.7**	**3.4**	23.0
Events [3]	9 (5.4%)	33.6 (0.4%)	18.6 (0.5%)	27.3 (1.2%)	23.6 (1.4%)	NA	NA

Fig. 3. Phase-based motion estimation for "Translation tree", "Yosemite", "Rubberwhale", "Grove2" (from Middlebury [2]), and "029_Brickbox2t2" (from UCL dataset [17]). The table reports the relative AEPE (in %) and the density of valid values (in % and in parenthesis) for standard optic flow methods and event-based methods. Highlighted values correspond to the lowest errors but, when comparing event-based methods we highlight multiple methods, if the error is similar but the density is significantly different.

Table 1. Relative AEPE (in %) and density of valid values (in % and in parenthesis) for event-based motion estimation methods

Sequence	Sun [22]	Phase events	Barranco [3]	Benosman [6]	Benosman [7]
Brickbox1	68.6 (15.5%)	**59.8** (15.5%)	73.8 (10.6%)	88.5 (10.0%)	88.8(17.9%)
Brickbox2	41.2 (32.7%)	**38.8** (32.7%)	68.5 (5.9%)	86.0 (31.7%)	71.7 (42.1%)
robot#1	NA	**44.2** (9.9%)	62 (10.3%)	88.7 (14.3%)	63.5 (2.5%)
robot#2	NA	**45.8** (8.9%)	66.2 (10.71%)	83 (3.4%)	60.5 (3%)
robot#3	NA	**42.3** (10.8%)	48.3 (8.75%)	80 (11.8%)	71 (9.4%)
robot#4	NA	**44.3** (7.11%)	51.2 (11.2%)	85.8 (12.2%)	77.1 (6.1%)

6.1 Classic Synthetic Scenes for Event-Based Data

Due to the lack of event-based benchmark datasets, we created simulated data to compare our method to state-of-the-art Computer Vision algorithms. We select the central frames of sequences from the Middlebury dataset [2]. Using the provided ground-truth, the ground truth motion is derived for very small time intervals: e.g., we simulate about 50000 frames for every two frames. This means that if conventional frames are obtained at 20 fps, our simulation provides 1000000 fps. Using the image motion for the short time intervals, interpolation gives us frame-by-frame the changes in intensity, from which we then derive the events and their timestamps at $1\mu s$ temporal resolution. Occluded and disoccluded regions are handled separately, since most of the conventional

benchmarks do not provide the image motion for occluded regions. If the scene in the region and the camera are static, we fill in the intensity values with the ones extracted from the previous frame (for disoccluded regions), or the next frame (for occluded regions). If there is motion, for a texture-less region we also assume the background motion of the neighboring region, otherwise, we discard it.

The relative Average End Point Error is used as a measure of the accuracy of the estimation. We simulated the events for the following scenes: "Diverging Tree", "Translation Tree", and "Yosemite". These are synthetic sequences simulating a moving camera approaching a tree image, moving to the right in front of the same tree, and a fly-through over the Yosemite Valley. We also used the the real sequences "Dimetrodon" and "Rubberwhale", which were taken by static camera with different moving objects and the two scenes "Urban2" and "Grove2" with stronger textures and global motion.

Fig. 3 compares the proposed method, our contour-based event method [3] and top frame-based algorithms, among them the method of Sun [22], which ranked the first in 2014 in [2]. Since we are estimating normal flow, to compare with the ground-truth, we project the actual flow onto the gradient direction computed by our event-based method. In the figure, the 'density' denotes the percentage of points with normal flow data for a method, out the total number of points (as in [4,5,9,23]), except the points for which there is no ground-truth available. Note that the error is reported for the exact same locations when comparing with the conventional frame-based techniques. Bold fonts highlight the lowest error in the comparison of the phase-based method to classical methods. Regarding to the contour-based method [3], it is included separately only for the sake of clarity. The error reported for this last method does correspond to different locations and thus, the densities are not the same.

Our phase-based method performs significantly better than many conventional algorithms, specifically [12,14,16,20,24]. The frame-based estimation provides dense estimates, but is bad at contours, where most of the events are coming from. The top method of Sun [22] used in the comparison, employs a number of sophisticated Computer Vision techniques, such as a hierarchical matching scheme for large motions, median filtering to reduce outliers, occlusions, etc. Even though it outperforms our method for four scenes, we are still doing better for the others. Conventional image-based algorithms perform better with global motion, since the flow changes smoothly along the scene, and this is precisely what many methods such as the one in Sun assume. Analyzing the sequences, our method outperforms Sun for the sequences with a strong texture component and with no global motion. The phase based method is better than the event-based method [3] for three scenes. But the methods should not be compared generally, as address different regions, which is reflected in the different densities.

Standard datasets are not well suited for evaluating event-driven methods since often they have carefully chosen small inter-frame distances, there is no data at occlusions, and there is no 3D information. Nevertheless, the average

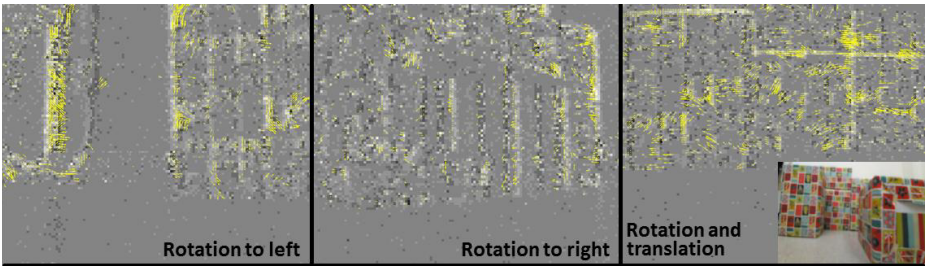


Fig. 4. Example of the scenes collected with the sensor mounted on a mobile platform. The image in the right shows a capture of the set up captured with a camera.

error for our phase-based method is 26.5 with 5.9% density, while the method in [3] achieves 22.4 but with only 1.78% valid values (3 times less).

6.2 Virtual Scenes for Event-Based Data

We created a dataset of synthetic sequences for evaluation of event-based methods. We used the sequences "026_Brickbox1t1", "029_Brickbox2t2" from the dataset [17]. These are two sequences with differently textured objects and with large inter-frame displacements. Using the know 3D models and 3D motion, we simulated the events to obtain our the ground-truth. Using this data we evaluated the accuracy of different different event-based algorithms [3,6,7] in Table 1. To allow for comparison to conventional techniques, the table also provides the results for the Sun method [22] . The motion estimation for the "029_Brickbox2t2" sequence is shown in Fig. 3.

From Table 1 we can infer that our method achieves better results in the relative AEPE for these high-frequency texture sequences. Our method performs even better than Sun's method, with an error reduction of 5 points (considering same density). In comparison to previous works our method achieves an average improvement of 30% with 4.1% more estimates. We use the same parameters described in the paper for [3], for [6] we set $th1 = 1000\mu s$ and $th2 = 0.25$, and for [7] we use $\Delta t = 500\mu s$ and 5 $\times$ 5 windows for the least-squares estimation. We assumed that the real sequences were recorded at 40 fps, and we simulated a temporal resolution of $10\mu s$ with 2500 samples between two consecutive frames.

6.3 Real Scenes for Event-Based Data

We also collected real data using the DVS camera on-board a mobile platform. We collected four sequences: "robot#1" and "robot#2" are pure zoom-in motion(translations for Z = 0.2 m/s); "robot#3" is a pure rotation (roll = 0, pitch = 0, yaw = $\pi/4$) and "robot#4" has both rotation (roll = 0, pitch = 0, yaw = $\pi/4$) and translation (Y = -0.15 m/s, Z = 0.5 m/s). The estimated motion is for 1s. The ground-truth data was derived from the odometry of the robot,

the sensor parameters (focal length and optic center), and depth measurement of the objects, which were positioned fronto-parallel to the sensor.

The setup consisting of three textured boxes at different depths, and the estimated motion for three sequences superimposed on the image of accumulated events are show in Fig. 4. As seen in Table 1, the algorithm presented in this paper achieves better results than any of the previous event-based methods in the literature. The difference is most significant for "robot#3", where due the rotation, the vertical components of the textures cause more events and reduce the accuracy of other methods. In these scenes the edges are not clean but appear broader caused by the textures in the background. This causes previous methods to fail while the method presented in the paper still achieves reasonable accuracy. The average error reduction in the worst case is about 13%, and in the best case about 40%, considering a similar number of image points with estimates.

7 Conclusions

Current event-based image motion estimation techniques have good performance at contours, where classical conventional methods fail. Using address events or a combination of event data with the intensity signal they have been shown to reduce computational complexity and perform at real time. However, existing methods rely on the assumption that close-by pixels from the same edge fire events also close in time, which is false for textured edges. In this paper we introduced an approach, that deals with such textured edges. We first presented a simple method for locating such textured edges. Second, we presented a method that uses the local phase of the event signal to accurately estimate image motion. Using the local phase instead of trying to reconstruct the intensity signals as in previous works, allows us to avoid the problem with textured edges.

We presented experimental results comparing the method to classic Computer Vision methods and other event-based approaches using synthetic sequences, and sequences collected with a sensor mounted on a mobile platform. Our method reduces the error in synthetic sequences, but it is hard to evaluate the significance the results. The results are more obvious for real scenes with textured objects, where the method was shown to decrease the error up to 30%.

We suggest that an optimal strategy, would be to use different techniques for image motion estimation, depending on the structure of data near a pixel. In future work we plan to develop a systematic method to decide for a given pixel, which method to use. Current event-based techniques are good for clean contours, as there they are accurate and have real-time performance, while our phase-based method requires extended time support for the filters, but has higher accuracy for textured contours.

Acknowledgments. The support of the European Union under the EU Marie Curie fellowship (FP7-PEOPLE-2012-IOF-33208) and the Cognitive Systems program (project POETICON++), the Junta de Andalucia VITVIR project (P11-TIC-8120), the National Science Foundation under INSPIRE grant SMA 1248056, and by DARPA through

U.S. Army grant W911NF-14-1-0384 under the Project: Shared Perception, Cognition and Reasoning for Autonomy are gratefully acknowledged.

References

1. Adelson, E.H., Bergen, J.R.: Spatiotemporal energy models for the perception of motion. Journal of Optical Society of America, A **2**(2), 284–299 (1985)
2. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. Journal Computer Vision **92**(1), 1–31 (2011)
3. Barranco, F., Fermuller, C., Aloimonos, Y.: Contour motion estimation for asynchronous event-driven cameras. Proc. of the IEEE **102**(10), 1537–1556 (2014)
4. Barranco, F., Tomasi, M., Diaz, J., Vanegas, M., Ros, E.: Parallel architecture for hierarchical optical flow estimation based on FPGA. IEEET on VLSI **20**(6), 1058–1067 (2012)
5. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. Int. Journal Computer Vision **12**, 43–77 (1994)
6. Benosman, R., Clercq, C., Lagorce, X., Ieng, S.H., Bartolozzi, C.: Event-based visual flow. IEEET Neural Networks Learning Systems **25**(2), 407–417 (2014)
7. Benosman, R., Ieng, S.H., Clercq, C., Bartolozzi, C., Srinivasan, M.: Asynchronous frameless event-based optical flow. Neural Networks **27**, 32–37 (2012)
8. Berner, R., Brandli, C., Yang, M., Liu, S.C., Delbruck, T.: A 240 x 180 10mw 12us latency sparse-output vision sensor for mobile applications. In: 2013 Symposium on VLSI Circuits (VLSIC), pp. C186–C187, June 2013
9. Brandt, J.: Improved accuracy in gradient-based optical flow estimation. Int. Journal of Computer Vision **25**(1), 5–22 (1997)
10. Brodsky, T., Fermüller, C., Aloimonos, Y.: Structure from motion: Beyond the epipolar constraint. Int. Journal Computer Vision **37**(3), 231–258 (2000)
11. Fermüller, C.: Passive navigation as a pattern recognition problem. Int. Journal of Computer Vision **14**(2), 147–158 (1995)
12. Fleet, D.J., Jepson, A.D.: Computation of component image velocity from local phase information. Int. Journal of Computer Vision **5**(1), 77–104 (1990)
13. Heeger, D.J.: Optical flow using spatiotemporal filters. Int. Journal of Computer Vision **1**(4), 279–302 (1988)
14. Horn, B.K.P., Schunck, B.G.: Determining optical flow. AI vol. 17, pp. 185–203 (1981)
15. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128x128 at 120db 15us latency asynchronous temporal contrast vision sensor. IEEE SSC **43**(2), 566–576 (2008)
16. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. Conf. on Artificial Intelligence **2**, 674–679 (1981)
17. Mac Aodha, O., Humayun, A., Pollefeys, M., Brostow, G.: Learning a confidence measure for optical flow. IEEET Pattern Analysis Machine Intelligence **35**(5), 1107–1120 (2013)
18. Mead, C.: Neuromorphic electronic systems. P. of IEEE **78**(10), 1629–1636 (1990)
19. Orfanidis, S.: Introduction to Signal Processing, Prentice Hall international editions. Prentice Hall (1996)
20. Otte, M., Nagel, H.H.: Optical flow estimation: Advances and comparisons. European Conference Computer Vision **800**, 49–60 (1994)
21. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: Computer Vision and Pattern Recognition, pp. 2432–2439 (2010)

22. Sun, D., Roth, S., Black, M.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. Int. Journal of Computer Vision **106**(2), 115–137 (2014)
23. Tomasi, M., Barranco, F., Vanegas, M., Díaz, J., Ros, E.: Fine grain pipeline architecture for high performance phase-based optical flow computation. Journal of Systems Architecture **56**(11), 577–587 (2010)
24. Uras, S., Girosi, F., Verri, A., Torre, V.: A computational approach to motion perception. Biological Cybernetics **60**(2), 79–87 (1988)
25. Watson, A.B.: Model of human visual-motion sensing. Journal of The Optical Society of America A-optics Image Science and Vision **2** (1985)

Transfer Learning

Domain Generalization Based on Transfer Component Analysis

Thomas Grubinger[1(✉)], Adriana Birlutiu[1,2], Holger Schöner[1], Thomas Natschläger[1], and Tom Heskes[3]

[1] Data Analysis Systems, Software Competence Center Hagenberg, Hagenberg, Austria
thomas.grubinger@scch.at
[2] Faculty of Science, 1 Decembrie 1918 University of Alba-Iulia, Alba Iulia, Romania
[3] Institute for Computing and Information Sciences, Radboud University Nijmegen, Nijmegen, The Netherlands

Abstract. This paper investigates domain generalization: How to use knowledge acquired from related domains and apply it to new domains? Transfer Component Analysis (TCA) learns a shared subspace by minimizing the dissimilarities across domains, while maximally preserving the data variance. We propose Multi-TCA, an extension of TCA to multiple domains as well as Multi-SSTCA, which is an extension of TCA for semi-supervised learning. In addition to the original application of TCA for domain adaptation problems, we show that Multi-TCA can also be applied for domain generalization. Multi-TCA and Multi-SSTCA are evaluated on two publicly available datasets with the tasks of landmine detection and Parkinson telemonitoring. Experimental results demonstrate that Multi-TCA can improve predictive performance on previously unseen domains.

1 Introduction

In many real-world applications one would like to make use of the knowledge acquired from related domains on previously unseen domains. This problem is known as *domain generalization*, and recently has started to gain attention in the machine learning community [3,12]. *Domain adaptation* [14] and domain generalization are subareas of transfer learning, aiming to find a shared subspace for related domains. While domain adaptation methods require at least some input data from the target domains, domain generalization methods are designed to generalize to previously unseen domains.

Most machine learning techniques rely on the assumption that the entire data, both training and testing, underlies the same data generation process. However, this assumption is often violated when data originates from multiple domains. Inequalities in the data generation process can lead to significant differences in marginal and conditional distributions of the data. Traditional machine learning methods can handle these differences only in two non-optimal ways: *i)* an individual model is fitted for each domain; a large amount of data is required,

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 325–334, 2015.
DOI: 10.1007/978-3-319-19258-1_28

which is expensive and the fitted models often do not generalize to new domains. *ii)* Differences in the data generation process are ignored, by learning a model on the pooled data. This approach usually results in low prediction accuracy and poor generalization [15].

Considerable effort has been made to remedy this problem (see [10,15] and references therein). Given one or more target domains, the idea of domain adaptation is to adapt a model trained on the training domains such that the generalization error on the test domains is minimized. The dissimilarities of data distributions from different domains are considered explicitly. Compared to a single model fit to all domains, predictive accuracy and the generalization to new domains can be improved. In comparison to tackling each domain independently, data can be used much more efficiently, as knowledge is transferred between domains. In this way, the effort to allocate data is massively reduced. The main drawback with this approach is that one has to re-train the models for every new target domain, which is time-consuming and inhibits real-time applications. Domain generalization is a solution to this problem: across-domain information is extracted from training data and can be used on the target domains without re-training.

The assumption in domain generalization is that the training and test domains are related. That is, there is at least some common information among the domains that is relevant for the considered machine learning task. Feature subsets of domain datasets can differ by a combination of various properties, including mean shift, scale, skewness, kurtosis, and rotation. For some applications such properties become obvious when doing exploratory data analysis. However, usually the description of the task and some domain knowledge already give strong indications to reject the hypothesis that the whole data is sampled from the same data generation process. For example, in medical applications, the data collected from two different patients usually cannot be assumed to be sampled from the same data generation process. See Section 2.2 and Section 4.1 for further examples.

Transfer Component Analysis (TCA) [14] is a domain adaptation technique that aims to learn a shared subspace between a source domain and a target domain. The shared subspace consists of some transfer components learned in a *reproducing kernel Hilbert space (RKHS)* [13] using *maximum mean discrepancy (MMD)* [5]. In the subspace spanned by these transfer components, data distributions of different domains should be close to each other and the task-relevant information of the original data should be preserved.

In this paper, we extend the formulation of TCA to multiple domains, compare it to the domain generalization method *Domain-Invariant Component Analysis (DICA)* Muandet *et al.*, and evaluate their benefits on real datasets. The same extension presented in this paper enables the use of TCA for *i)* domain adaptation problems with multiple domains; *ii)* domain generalization problems with multiple source and target domains. This paper focuses on the domain generalization setting. Our solution is based on the idea of learning a shared subspace between source domains and using this subspace for related target domains – without re-training. We present and evaluate two variants of our extension, an

unsupervised version to which we refer as *Multiple-Domain Transfer Component Analysis (Multi-TCA)* and a semi-supervised version called *Multiple-Domain Semi-Supervised Transfer Component Analysis (Multi-SSTCA)*.

The remainder of this paper is organized as follows: Section 2 discusses related work in domain generalization and the bigger area of transfer learning and domain adaptation. Section 3 presents our proposed extension to TCA in both supervised and unsupervised settings. Section 4 shows an experimental evaluation on two publicly available datasets from the UCI repository. Section 5 gives the conclusions and directions for future work.

2 Related Work

2.1 Domain Generalization

Although, there is a large amount of publications in the field of transfer learning and domain adaptation, very few studies address domain generalization. Recently, Muandet *et al.* [12] presented a method called *Domain-Invariant Component Analysis (DICA)*, which addresses the problem of domain generalization. DICA and its unsupervised version UDICA are closely related to Multi-SSTCA and Multi-TCA. UDICA and Multi-TCA are derived differently but have similar objectives. They both try to find a subspace where: *i)* the distance between the domain datasets is minimized; *ii)* the variance in the feature space is maximized. DICA is an extension of UDICA that takes the functional relationship between X and Y into account – the derivation is again different to the extension of Multi-TCA to Multi-SSTCA. Besides the different derivation, Multi-SSTCA is more versatile than DICA as *i)* Multi-SSTCA can also consider the *manifold information* (see objective 3 in Section 3.2); *ii)* the definition of Multi-SSTCA can handle missing class labels, allowing the application of Multi-SSTCA to semi-supervised domain generalization and domain adaptation tasks.

Persello and Bruzzone [17] address domain generalization by selecting features that minimize the shift in the domain dataset distributions. Their selection criteria select variables that have *i)* high dependency with the target variable and *ii)* invariant data distributions across domains.

2.2 Transfer Learning and Domain Adaptation

In contrast to domain generalization, transfer learning and domain adaptation have received a lot of attention in the recent years. Transfer learning [15] aims at transferring knowledge from some previous tasks to a target task when the latter has limited training data. Domain adaptation [2,10,18] can be viewed as a subdomain of transfer learning, that deals primarily with a mismatch between training and test input distributions. A popular and intuitive approach for domain adaptation is to make the source and target distributions as similar as possible. The methods that follow this line of research can be grouped into two categories. Firstly, sample re-weighting [4,8] approaches, which apply weights to the source

samples to adjust their influence in the source distribution. Secondly, learning a shared subspace is a common approach in settings where there is distribution mismatch. A typical approach in multi-task learning is to uncover a latent feature space that is shared across tasks. Commonly, latent factors are designed to represent statistical properties and/or the geometric structure of the data. Methods of this category exist for problems with different feature spaces [21] or marginal distributions [11].

There are quite some successful applications of transfer learning methods in different real-world applications. One application is a WiFi-based indoor localization problem presented in [16]. In this application the data is highly dependent on time, space and the client device. Transfer learning was successfully applied to transfer localization models over these dependencies. Another application area is in the field of image processing, e.g. Hinton *et al.* [7] apply transfer learning in a face recognition and a handwritten digit example. In [12,20], Varnek *et al.* apply transfer learning techniques in biological applications.

3 Transfer Component Analysis for Domain Generalization

TCA aims to learn a good feature representation across different distributions, i.e., a shared subspace. In the learned subspace the distance of the individual dataset distributions is minimized and properties of the data are preserved. The use of a RKHS provides the possibility to use non-linear kernels. Subsequently, any machine learning method for regression, classification or clustering can be used on the identified subspace.

TCA has originally been designed to work with the most common transfer learning setting. In this setting, the goal is to find a common representation for one source domain $\mathcal{D}_S$ and one target domain $\mathcal{D}_T$, with at least some input data X_S, X_T existing in both domains. Here, a kernel-induced feature map ϕ is learned from $\{X_S, X_T\}$. Once transformed, the combined source and target data can be used in the subsequent machine learning task. The TCA algorithm and the learning setting described in this paper are different to the original Paper presented by Pan *et al.* [14] in the following aspects:

- **Differences in the learning algorithm:** This paper gives an extension of TCA to more than two domains. This can simply be achieved by extending the cost, weight and kernel matrices – see Equation 1 for an extension of the cost function as well as Equation 2 and Equation 3 for extension of the matrices.
- **Differences in the learning task:** The original paper considers two domains with input data from both domains. However, in our application, the TCA transformation is applied to domains without any input data. Here, first a common subspace for the source domain datasets $X_1, ..., X_S$ is learned. The learned model can then be applied to the target domain datasets $X_{S+1}, ..., X_U$. The assumption is that the common data properties extracted from the source datasets also apply to the target data.

The reminder of this paper describes the extensions from TCA/SSTCA to Multi-TCA/Multi-SSTCA. See Pan *et al.* [14], and references within, for a more detailed description of TCA/SSTCA, especially for the derivation of formulas.

3.1 (Unsupervised) Transfer Component Analysis

Multi-TCA is applicable if $P(X_s) \neq P(X_u), 1 \leq s < u \leq U$, where X_s, X_u are domain datasets, $P(X_s)$ is the probability distribution of X_s and U is the total number of source and target domain datasets. The goal of Multi-TCA is to find a feature map ϕ such that $P(\phi(X_s)) \approx P(\phi(X_u))$.

Assume ϕ is a feature map induced by a universal kernel. *Maximum mean discrepancy (MMD)* [5] measures the distance between the empirical means of two domains in the RKHS. We extend to more than two domains

$$\text{MMD} = \frac{1}{S}\sum_{s=1}^{S} ||\mu_{x_s} - \mu_{\bar{x}}||^2_{\mathcal{H}}. \tag{1}$$

Here, $\mu_{x_s} = \frac{1}{n_s}\sum_{i=1}^{n_s}\phi(x_{si})$ and $\mu_{\bar{x}} = \frac{1}{S}\sum_{s=1}^{S}\mu_{x_s}$, where n_s are the number of instances from X_s. S is the number of training domain datasets, x_{s_i} denotes the i^{th} instance of X_s and $||\cdot||_{\mathcal{H}}$ is the RKHS norm.

Let K be a combined Gram matrix [19] of the cross-domain data of the training domain $X_1, X_2, ..., X_S$:

$$K = \begin{bmatrix} K_{X_1,X_1} & K_{X_1,X_2} & \dots & K_{X_1,X_S} \\ K_{X_2,X_1} & K_{X_2,X_2} & \dots & K_{X_2,X_S} \\ \vdots & \vdots & \ddots & \vdots \\ K_{X_S,X_1} & K_{X_S,X_2} & \dots & K_{X_S,X_S} \end{bmatrix} \in \mathbb{R}^{N\times N} \tag{2}$$

where $N = \sum_{s=1}^{S} n_s$. Each element $K_{i,j}$ of K is given by $\phi(x_i)^T\phi(x_j)$. The calculation of MMD in Equation 1 can be rewritten as $tr(KL)$, where $L_{i,j}$ is defined as

$$L_{i,j} = \begin{cases} \frac{S-1}{N^2 n_s^2} & \text{if } x_i, x_j \in X_s \\ -\frac{1}{N^2 n_s n_u} & \text{if } x_i \in X_s, x_j \in X_u \text{ and } s \neq u \end{cases} \tag{3}$$

and $s, u \in \{1, ..., S\}$. The computational expensive semi-definite programming can be avoided by the use of a parametric kernel map $K = (KK^{-1/2})(K^{-1/2}K)$. Pan *et al.* [14] shows that the resulting kernel matrix $\tilde{K} = KWW^TK$, where $W \in \mathbb{R}^{N\times m}$, $m \ll N$ is an orthogonal transformation matrix that is found by Multi-TCA. As a result the MMD distance in Equation 1 can be rewritten as

$$\text{MMD} = tr((KWW^TK)L) = tr(W^TKLKW). \tag{4}$$

Similarly to PCA and KPCA [19], the second objective of Multi-TCA is to maximally preserve the data variance. The variance of the projected samples is W^TKHKW, where centering matrix H is defined as $H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Here,

$\mathbf{1} \in \mathbb{R}^N$ is a column vector with all ones and $I \in \mathbb{R}^{N \times N}$ is the identity matrix. With a regulation term $tr(W^T W)$ and the tradeoff parameter μ, the objective of Multi-TCA can be formulated as

$$\min_W tr(W^T KLKW) + \mu\ tr(W^T W),\ \text{s.t. } W^T KHKW = I. \tag{5}$$

The embedding of the data in the latent space is given by $W^T K$. The solution of W is given by the $m \ll N$ leading eigenvectors of

$$(KLK + \mu I)^{-1} KHK, \tag{6}$$

where $\mu > 0$ is a tradeoff parameter that is usually needed to control the complexity of W.

3.2 Semi-Supervised Transfer Component Analysis

Multi-SSTCA is an extension to Multi-TCA based on SSTCA from Pan *et al.* [14] that also considers the conditional probabilities $P(Y_i|X_i)$, $i \in 1, ..., S$ and optimizes the following three objectives:

1. *Distribution Matching:* as in Multi-TCA, the first objective is to minimize the distribution differences – measured by the MMD criterion (Equation 1) – between the domain datasets.
2. *Label Dependence:* maximize the dependency between the embedding and the labels. This is achieved by the use of the Hilbert-Schmidt Independence Criterion (HSIC) [6] given by $\max_{K \succeq 0} tr(HKHK_{yy})$, where $K_{yy} = \gamma_w K_l + (1 - \gamma_w) K_v$. Here, $k_l = \phi(y_i, y_j)$, $K_v = I$ and γ_w is a tradeoff parameter that balances the label dependence with the data variance terms. The second objective is to
$$\max_W tr(W^T KHK_{yy} HKW). \tag{7}$$
3. *Locality Preserving:* Multi-SSTCA uses the manifold regularization of Belkin *et al.* [1]. In order to preserve locality, each x_i and x_j that are neighbors in the input space should also be neighbors in the data's embedding. A matrix $M \in \mathbb{R}^{N \times N}$ is constructed by $M_{i,j} = \exp(-(x_i - x_j)^2 / 2\sigma^2)$ if x_i is one of the k nearest neighbors of x_j, and $M_{i,j} = 0$ otherwise. The graph Laplacian is defined by $A = D - M$, where $D \in \mathbb{R}^{N \times N}$ is a diagonal matrix with entries $D_{i,i} = \sum_{j=1}^{N} M_{i,j}$. The third objective is to
$$\min_W \sum_{(i,j) \in N} M_{i,j} ||[W^T K]_i - [W^T K]_j||^2 = tr(W^T KAKW). \tag{8}$$

For Multi-SSTCA, the objective function is

$$\begin{aligned} &\min_W tr(W^T KLKW) + \frac{\lambda}{n^2} tr(W^T KAKW) + \mu\ tr(W^T W) \\ &\quad \text{s.t. } W^T KHK_{yy} HKW = I \end{aligned} \tag{9}$$

and the solution of W is given by the $m < N$ leading eigenvectors of

$$(K(L + \lambda A)K + \mu I)^{-1} KHK_{yy}HK. \tag{10}$$

Note that Multi-SSTCA is a semi-supervised method in the domain adaptation setting, where the input data from the target domain is used without the target data. In domain generalization problems no target data is used at all. Thus, in domain generalization the special case occurs where Multi-SSTCA is used in a supervised setting – provided that no labels are missing in the source domains. Despite of this technicality, the same method can be used for both problem settings, domain adaptation and domain generalization.

4 Experimental Evaluation

4.1 Experimental Setup

We use two datasets for the experimental evaluation. *i)* The landmine data represents a landmine detection problem [22], based on airborne synthetic-aperture radar measurements. It has 9 features and 29 domains. As the class labels (1 for landmine and 0 for clutter) are highly unbalanced, we took all instances with class 1 and randomly selected the same amount of class 0 examples in each repetition, resulting in a total number of 1808 instances. *ii)* The Parkinson telemonitoring dataset [9], which consists of biomedical voice measurements from 42 people with early-stage Parkinson's disease. The Parkinson data was collected in a six-month trial of a telemonitoring device for remote symptom progression monitoring (5875 recordings in total). The goal is to predict the clinician's scoring of Parkinson's disease symptom based on 16 voice measurements. There are two scores to predict: the motor score and the total score on the Unified Parkinson's Disease Rating Scale (UPDRS). We consider each dataset, related to one patient, as a domain.

We compared Multi-TCA and Multi-SSTCA as preprocessor for a linear SVM with: *i)* KPCA with an RBF kernel as preprocessor for a linear SVM; *ii)* an SVM with a linear kernel and an SVM with an RBF kernel without any preprocessing. For the landmine data 5 training domains are selected from each of *relatively highly foliated* (domains $1 - 15$) and *bare earth or desert* (domains $16 - 29$) regions. For the Parkinson data we also consider 10 training domains. For both datasets, the remaining domains are used for testing. We randomly repeat 25 times the selection of training and testing domains. Parameters are selected by 5-folds cross-validation.

On both datasets, the number of components for all preprocessors is selected from $\{1..15\}$. For the input data we use an RBF kernel and select $\gamma \in \{0.005..0.5\}$ and $\gamma \in \{0.1..1\}$ for the Parkinson data and the landmine data, respectively. For classification with DICA and Multi-SSTCA we apply the output kernel $k_{yy}(y_i, y_j) = 1$ if $y_i = y_j$ and -1 otherwise. For regression we use an RBF kernel with $\gamma = 0.1$. We set the Multi-TCA/Multi-SSTCA parameter $\mu = 0.1$ and the UDICA/DICA parameter $\lambda = 0.1$ for the landmine data. For the Parkinson data

$\mu = 0.01$ and $\lambda = 0.01$. For UDICA and DICA $\epsilon = 0.0001$. The Multi-SSTCA parameter $\gamma_w = 0.5$. For Multi-SSTCA we build one model considering the manifold information ($\lambda = 1000$) and one without considering manifold information ($\lambda = 0$). We construct A using an RBF kernel ($\gamma = 1$) and 4-nearest neighbors. For SVM we select $C \in \{10^{-4}..10^{4}\}$ and for γ we apply the same ranges that are used by the preprocessors.

4.2 Experimental Results

The relative performance of the algorithms are summarized in Table 1. Performance on the test data is measured by misclassification rate (MC) for the landmine data and root mean square error (RMSE) for the Parkinson data.

The results in Table 1 show that Multi-TCA perform best on the landmine data, followed by UDICA. DICA performs best on the Motor score Parkinson problems, closely followed by the performance of Multi-SSTCA. With the same RMSE of 8.73, Multi-SSTCA and DICA are also the best methods on the Total score Parkinson problem. While taking the labels into account is clearly beneficial on the Parkinson data, it is not on the landmine data, where Multi-SSTCA and DICA perform worse than their unsupervised versions and KPCA. For Multi-SSTCA and the evaluated datasets, the modeling of the manifold information does not lead to any considerable improvements.

Table 1. Predictive performances $mean(std)$ of the evaluated methods

Preprocessor	SVM Kernel	Parkinson Motor score	Parkinson Total score	Landmine
Multi-TCA	linear	**32.39 ± 1.49**	11.39 ± 0.76	8.91 ± 0.69
Multi-SSTCA($\lambda = 0$)	linear	32.81 ± 1.32	11.30 ± 0.83	**8.73 ± 0.77**
Multi-SSTCA($\lambda = 1000$)	linear	33.14 ± 1.61	11.29 ± 0.82	8.76 ± 0.76
UDICA	linear	32.43 ± 1.26	11.58 ± 0.80	9.02 ± 0.68
DICA	linear	33.56 ± 1.20	**11.25 ± 0.82**	**8.73 ± 0.75**
KPCA	linear	32.71 ± 1.53	11.53 ± 0.85	8.89 ± 0.70
None	linear	32.66 ± 1.43	12.30 ± 3.90	9.15 ± 1.27
None	RBF	32.51 ± 1.19	11.56 ± 1.28	9.02 ± 0.98

On the evaluated datasets, differences in computation time can be neglected for the domain generalization methods. For example, the required computation time of the considered methods (Multi-TCA, Multi-SSTCA, UDICA, DICA) on the Parkinson Motor Score data are all within 32 ± 1 seconds on a standard PC. The most computational demanding tasks are the eigenvalue decomposition followed by the computation of kernel matrix K (Equation 2), which are required by each of the methods.

5 Conclusion and Future Work

In this paper we presented an extension of TCA to multiple domains and successfully applied it for domain generalization. We showed that improvements in predictive performance can be achieved by aligning related datasets via the domain

generalization methods Multi-(SS)TCA and (U)DICA. The performances of Multi-TCA and Multi-SSTCA on the two benchmark datasets were comparable to the performances of UDICA and DICA, respectively. However, compared to DICA, Multi-SSTCA can also take the manifold information into account (locality preserving) and is applicable for semi-supervised domain generalization tasks and domain adaptation. Although, the manifold information did not lead to any considerable improvement on the evaluated datasets, Pan *et al.* [14] showed its usefulness on simulated and real world domain adaptation problems. In the future we want to investigate the usefulness of the manifold information for domain generalization problems. Multi-TCA/Multi-SSTCA has many parameters that can be optimized. We plan to conduct sensitivity analysis and work on a parameter selection strategy.

Acknowledgments. Parts of this work were carried out in mpcEnergy, a project supported within the program Regionale Wettbewerbsfähigkeit OÖ 2007-2013 by the European Fund for Regional Development as well as the State of Upper Austria.

References

1. Belkin, M.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J. Mach. Learn. Res. **7**, 2399–2434 (2006)
2. Bickel, S., Brückner, M., Scheffer, T.: Discriminative learning under covariate shift. J. Mach. Learn. Res. **10**, 2137–2155 (2009)
3. Blanchard, G., Lee, G., Scott, C.: Generalizing from several related classification tasks to a new unlabeled sample. In: NIPS, pp. 2178–2186 (2011)
4. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML, pp. 222–230 (2013)
5. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. In: NIPS, pp. 513–520 (2006)
6. Gretton, A., Bousquet, O., Smola, A.J., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005)
7. Hinton, G.E., Salakhutdinov, R.: Using deep belief nets to learn covariance kernels for gaussian processes. In: NIPS, pp. 1249–1256 (2007)
8. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS, pp. 601–608 (2006)
9. Little, M., McSharry, P., Roberts, S., Costello, D., Moroz, I.: Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. BioMedical Engineering OnLine **6:23**(1) (2007)
10. Long, M., Pan, S.J., St Yu, P., Wang, J., Ding, G.: Adaptation regularization: A general framework for transfer learning. IEEE Trans. on Know. and Data Eng. **26**(5), 1076–1089 (2014)
11. Long, M., Wang, J., Ding, G., Shen, D., Yang, Q.: Transfer learning with graph co-regularization. In: Proc. of the 26th Conf. on Art. Int., pp. 1805–1818. AAAI (2012)
12. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: Proc. of the 30th Int. Conf. on Mach. Learn., pp. 10–18 (2013)

13. Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. on Neural Networks **12**(2), 181–201 (2001)
14. Pan, S.J., Tsang, I., Kwok, J., Yang, Q.: Domain adaptation via transfer component analysis. Trans. on Neural Networks **22**(2), 199–210 (2011)
15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Know. and Data Eng. **22**(10), 1345–1359 (2010)
16. Pan, S.J., Zheng, V.W., Yang, Q., Hu, D.H.: Transfer learning for wifi-based indoor localization. In Proc. of the Workshop on Trans. Learn. for Complex Tasks, of the 23rd AAAI Conf. on Art. Int., pp. 43–48 (2008)
17. Persello, C., Bruzzone, L.: Relevant and invariant feature selection of hyperspectral images for domain generalization. In: Int. Geoscience and Remote Sensing Symposium (IGARSS), pp. 3562–3565. IEEE (2014)
18. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press (2009)
19. Schölkopf, B., Smola, A., Müller, K.: Kernel principal component analysis (1999)
20. Varnek, A., Gaudin, C., Marcou, G., Baskin, I., Pandey, A.K., Tetko, I.V.: Inductive transfer of knowledge: application of multi-task learning and feature net approaches to model tissue-air partition coefficients. J. of Chem. Inf. and Modeling **49**(1), 133–144 (2009)
21. Wang, C., Mahadevan, S.: Heterogeneous domain adaptation using manifold alignment. In: Proc. of The Twenty-Second Int. Joint Conf. on Art. Int., vol. 2, pp. 1541–1546. AAAI (2011)
22. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multitask learning for classication with Dirichlet process priors. J. Mach. Learn. Res. **35**(8), 35–63 (2007)

Deep Transfer Learning Ensemble for Classification

Chetak Kandaswamy[1,2,3(✉)], Luís M. Silva[2,4], Luís A. Alexandre[5], and Jorge M. Santos[2,6]

[1] Instituto de Investigação e Inovação em Saúde, Universidade do Porto, Porto, Portugal
[2] INEB - Instituto de Engenharia Biomédica, Porto, Portugal
[3] Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Porto, Portugal
[4] Departamento de Matemática, Universidade de Aveiro, Aveiro, Portugal
[5] Universidade da Beira Interior & Instituto de Telecomunicações, Covilhã, Portugal
[6] Instituto Superior de Engenharia, Politécnico do Porto, Porto, Portugal

Abstract. Transfer learning algorithms typically assume that the training data and the test data come from different distribution. It is better at adapting to learn new tasks and concepts more quickly and accurately by exploiting previously gained knowledge. Deep Transfer Learning (DTL) emerged as a new paradigm in transfer learning in which a deep model offer greater flexibility in extracting high-level features. DTL offers selective layer based transference, and it is problem specific. In this paper, we propose the Ensemble of Deep Transfer Learning (EDTL) methodology to reduce the impact of selective layer based transference and provide optimized framework to work for three major transfer learning cases. Empirical results on character, object and biomedical image recognition tasks achieves that the proposed method indicate statistically significant classification accuracy over the other established transfer learning method.

Keywords: Deep learning · Transfer learning · Ensemble

1 Introduction

Transfer of learning is a well established concept in many fields, including machine learning. Transfer learning (TL) approach in machine learning refers to the procedure employed to train a source model and then transfer the knowledge (learning) across different problems. In principle, it gives better generalization with less computational effort even when the training has a considerable amount of unlabelled data. TL offers several advantages over traditional machine learning specially for non-stationary environments where the training and test samples may be drawn from different marginal distributions or the classification tasks may not be identical.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 335–348, 2015.
DOI: 10.1007/978-3-319-19258-1_29

Several viable solutions have appeared in the literature to train machines for non-stationary environments in the past two decades: in lifelong learning [1], where it is assumed that the learner faces an entire family of learning tasks, not just a single one; multi-task learning [2], where it is possible to learn the tasks simultaneously; cross-domain learning [3], where it is possible to reuse the learning when the distributions are correlated; self-taught learning [4], where the machine makes use of large number of easily available unlabelled data to build high-level representations to use for supervised classification tasks.

Recent developments in neural networks inspired by the biological structure of the visual cortex like convolutional networks (CNN) [5], deep belief nets [6], and Stacked denoising autoencoders [7] [8] combined with faster processing computational capabilities lead to the development of deep learning. Deep learning models extract useful information from the input data, constructing multiple levels of representation or learning a hierarchy of features. It potentially leads to progressively more abstract features at higher layers. It is observed that the bottom-layer features are standard regardless of the cost function or dataset used, called as *general,* while the top-layer features depend greatly on the chosen dataset and task, so called as *specific* [9] [11] [12].

The transference of hierarchical features obtained by deep learning for solving classification task, lead to the emergence of Deep Transfer Learning (DTL). DTL an approach in which a deep model is trained on a source problem, and then reused to solve a target problem. In the case of DTL, transference occurs due to two reasons: 1) transferring supervised or unsupervised features from the source problem [9] and 2) retraining only unlocked layers of the network by constraining not to over train for the target task [11].

The DTL approach has proven to be successful in many object and image recognition problems using a layer-by-layer feature transference on large-scale data by transferring hidden layers [10] and retrain unlocked layers [11] [12]. Transferring features of convolutional neural network trained on ILSVRC dataset and retrained to solve several visual classification task on various datasets perform better than state-of-the-art methods [13].

All these above DTL methods have shown that there is a limitation on choosing the various selective layer based transference conditions to solve the new target problem. They do not tackle the problem of negative feature transference and also ambiguous in selecting the layers to be transferred and retrained for the target task.

We may therefore pose the following question:

– Will the learning algorithm be able to adapt by combining the outputs of various selective layer based transference conditions of the deep learning model, in a way, non-negativity constraints as required by the feature transference are no longer needed?

It would be interesting to combine the outputs using ensemble methods. Following the rather standard ensemble methods with deep learning models for classification task. The recent ensemble of deep models using different initial weights on a multi-column CNN in [16], or using combination of different deep

learning methods like CNN and recurrent neural networks in [14], or using combination of shallow and deep models in [15] have shown significant improvement in accuracy. All these ensemble of deep models have shown increased storage and computation.

In this paper, we propose the Ensemble of Deep Transfer Learning (EDTL) approach which combines the advantage of using ensemble of deep models and deep transfer learning. We hypothesise the ensemble of various selective layer based transference on deep models removes non-negativity constraints and speeds up the computation. We study its two conditions of feature transference: 1) transfer specific features, and 2) retrain specific features (splitting of co-adapted neurons). As we will see, EDTL not only effectively reduces the issue of selective layer-based transference as well as improves performance over the established positive transference situations.

We reduce the transferability gap, the gap between the performance of the transference versus the no transference approach increases proportionally to the distance between the source and target distributions [3]. In order to distinguish how different the target distribution is from the source distribution, we use Jensen-Shannon divergence [17] as a metric to measure the degree of heterogeneity between distributions

2 Notations and Problem Settings

Let's represent a dataset by a set of tuples $D = (x_n, y_n) \in X \times Y$, where X is the input space and Y is a set of labels. Assume that the n instances are drawn by a sampling process from the input space X with a certain probability distribution $P(X)$. The dataset is split into subsets of training, validation and test drawn from the same distribution $P(X)$. We assume that the "source" dataset D_S with input space X_S and a set of labels Y_S is drawn from a distribution $P_S(X)$ and the "target" dataset D_T with input space X_T and a set of labels Y_T is drawn from a distribution $P_T(X)$. Such $P_S(X)$ and $P_T(X)$ may be equal or different.

Traditionally, the goal of transfer learning is to transfer the learning (knowledge) from a source problem input space X_S to one or more problems, or distributions to efficiently develop an effective hypothesis for a new task, problem, or distribution [3]. In supervised learning problems, the source and target marginal distribution and classification tasks may be equal or different. In this framework of transfer learning, four possible cases of transfer learning problems can be identified:

We use the well known Jensen-Shannon divergence (JSD) [17] as a measure to compute the difference between two datasets distribution and is given by:

$$D_{JS}(p||q) = \alpha D_{KL}(p||r) + \beta D_{KL}(q||r), \quad \text{with } r = \alpha p + \beta q \tag{1}$$

where D_{KL} is the Kullback-Leibler divergence.

When $\alpha = \beta = 1/2$ in eq.2 we are dealing with the *specific* Jensen-Shannon divergence and D_{JS} is lower- and upper-bounded by 0 and 1, respectively, when using logarithm base 2 [17]. This means that when $D_{JS}(p||q) = 0$ we can consider that p and q are identical and when $D_{JS}(p||q) = 1$, the distributions are different.

Table 1. Transfer Learning cases

Distribution		
Marginal	Labels	Case
$P_S(X) = P_T(X)$	$Y_S = Y_T$	no transfer learning
$P_S(X) \neq P_T(X)$	$Y_S = Y_T$	I
$P_S(X) = P_T(X)$	$Y_S \neq Y_T$	II
$P_S(X) \neq P_T(X)$	$Y_S \neq Y_T$	III

2.1 Established Frameworks

Baseline (BL): Stacked Denoising Autoencoders (SDA) are multiple layer networks where each one is trained as a denoising autoencoder (dA) (see Fig. 1-BL). SDA training comprises of two stages: an unsupervised pre-training stage followed by a supervised fine-tuning stage. During pre-training (PT), the network is generated by stacking multiple dA one on top of each other thus learning *unsupervised features*, represented as a vector $U(w)$ of optimal weights and biases. Then, a logistic regression layer is added on top and the whole network and fine-tuned in a supervised way, thus learning *supervised features* $\mathbf{w} = (w^1, ..., w^K)$, where K is the number of layers.

Transfer Learning (TL): We first train the source network with the source data D_S and Y_S and then copy its hidden layers to the target network. In case $Y_S \neq Y_T$, then we add a classifier layer randomly initialized. The network is trained towards the target task Y_T. If the performance of the newly trained target network exceeds the performance of the baseline approach we have positive transference; otherwise we have negative transference.

Transferred Layers: We select a particular layer or set of layers of the whole baseline network to transfer. For example we may select to transfer first layer features of the baseline approach to the target network, that is, $w_S^1 \Rightarrow w_T^1$. The rest of the target network layer features are randomly initialized.

Retraining Layers: Once the features are transferred to the target network, we add a logistic regression layer for the target task Y_T. We have a choice to fine-tune this entire network $\mathbf{w}_T$ as a multi-layer perceptron using back-propagation or *lock* a layer[11] [12], meaning the transferred feature from source network $\mathbf{w_S^1} \Rightarrow \mathbf{w_T^1}$ do not change during the error propagation for the target task. Thus giving a choice of whether or not to fine-tune the certain layers of the target network. This opens up several possible approaches to solve a problem as shown in Fig .1, TLu and TLs, where the layers are optionally locked or unlocked. This causes fragile co-adaptation of neurons between layers leading to optimization difficulties. The choice of whether or not to fine-tune the first layer of the target network depends on the size of the target dataset and number of parameters [10]. [13]

Transfer Learning Unsupervised (TLu): We transfer the unsupervised features of the SDA model from the source to the target network, i.e., $U(\mathbf{w}_S) \Rightarrow \mathbf{w}_T$ as depicted in Fig. 1, TLu. Once the features are transferred to the target network, we add a logistic regression layer for the target task Y_T. Then we fine-tune the entire classifier like a regular multi-layer perceptron with back-propagation choosing to lock or unlock certain layers to solve the target task.

Transfer Learning Supervised (TLs): The trained weights of the BL approach are used. For example we transfer features from source to target network, i.e., $\mathbf{w}_S \Rightarrow \mathbf{w}_T$. Then we back-propagate choosing to lock or unlock certain layers to solve the target task as illustrated in Fig. 1 TLs.

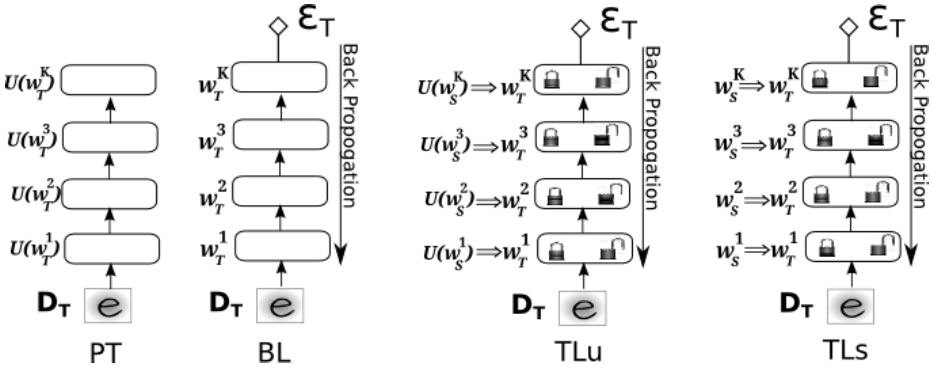


Fig. 1. A pictorial representation of approaches: Pre-training (PT), Baseline (BL), Transfer Learning unsupervised (TLu), and Transfer Learning supervised (TLs).

3 Ensemble of Deep Transfer Learning Framework

In this paper we propose an Ensemble of Deep Transfer Learning (EDTL) where we combine the main advantage of deep transfer learning with the traditional ensemble learning. The DTL offers the knowledge (features) learnt in a source domain providing a good initialization for the learning task in a target problem, better than starting the learning in the target domain at random with possibility of transferring generic features. In here we propose to ensemble the various DTL models between both domains. The intuition is that, like in traditional ensemble, train a DTL model with various transfer and retrain conditions and combine their outputs, treating them as a *committee* of decision makers. Numerous empirical and theoretical studies have demonstrated that ensemble (committee) models often obtain higher accuracy than single models [20].

The overall framework of EDTL depicted in Fig. 2, employs a deep model learnt on the source domain and apply DTL with various conditions, i.e., transfer hidden layer (transfer or randomly initialize) and then retrain (lock or unlock) the network to the target domain. Compute posterior probabilities $P_T(y|x)$ each

Algorithm 1. Pseudocode for baseline and Transfer learning approach

Baseline: Initialize randomly:
Given a two datasets D_A and D_B, Select a dataset and train the network with input x
{**Stage 1:** Pretrain the Network}
build SDA by greedy layer
for k in number of hidden layers **do**
  randomly initialize: W_k
  {Build denoising autoencoder (dA)}
  for each epoch in Pretraining **do**
    Corrupt the input, $x = x + noise$
    $hidden\,layer = Sigmoid(W_k x + bias)$
    $reconstruct = Sigmoid(W'_k x + bias')$
    minimize cross-entropy loss and update weight vector
  end for
  stack the dA's
end for
{**Stage 2:** Fine-tune the Network}
add a logistic regression layer with Y labels
for each epoch in Fine-tuining **do**
  backpropogate the errors
  update the weights
  calculate validation error on validation set
  if best validation error < validation error **then**
    update weights of the network
    best validation error = validation error
    calculate test error on test set
    best test error = current error
  end if
end for
error = best test error

Initialize with trained features D_A:
Given a two datasets D_A and D_B, with tasks Y_A and Y_B,
Select D_A dataset and train the network A as described on the left side.
{**Stage 1:** Transfer the features}
Select a reuse mode: TLu or and TLs
Select which hidden layers to transfer
if $Y_A \neq Y_B$ **then**
  chop of the logistic layer
end if
for k in number of layers **do**
  if layer = transfer **then**
    if mode = TLu **then**
      transfer unsupervised features
      $U(w_A^k) \Rightarrow w_B^k$
    else if mode = TLs **then**
      transfer supervised features
      $w_A^k \Rightarrow w_B^k$
    end if
  else if layer = no transfer **then**
    randomly initialize weights w_B^k
  end if
end for
{**Stage 2:** Fine-tune the Network}
if $Y_A = Y_B$ **then**
  add a logistic regression layer with Y_B labels
end if
for each epoch in Fine-tuining **do**
  backpropogate the errors
  if lock is TRUE in each Layer **then**
    no update of weights
  else
    update the weights
  end if
  calculate validation error on validation set
  if best validation error < validation error **then**
    update weights of the network
    best validation error = validation error
    calculate test error on test set
    best test error = current error
  end if
end for
error = best test error

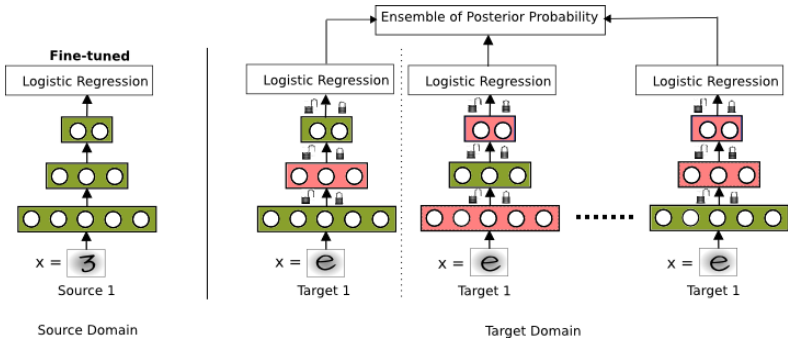


Fig. 2. A pictorial representation of Ensemble of Deep Transfer learning

of the DTL model for target task. Then obtain class-probabilities using ensemble the posterior probabilities $P_T(y|x)$ of each model. The model are trained on baseline method (BL) using standard deep learning approach and the deep transfer learning approach process as listed in Algorithm 1.

The bottom-layer features, called as *general,* similarly the top-layer features, called as *specific*. The pseudo-code for the EDTL process is listed in Algorithm 2, study the two conditions of feature transference: 1) transfer specific features, and 2) retrain specific features (splitting of co-adapted neurons, meaning fragile co-adaptation of neurons is broken by splitting of transferred layer and randomly initialized layer leads to difficulty in optimization [11]).

Algorithm 2. Pseudocode for EDTL

Initialize with trained features D_S**:**
Given two datasets D_S and D_T, with tasks Y_S and Y_T, drawn from P_S and P_T distributions.
Let the total number of models in the ensemble be M
{Select type of TL interaction to evaluate}
if evaluate == co-adapted interactions **then**
    M = possible combination of retrained layers
else if evaluate == generic vs. specific **then**
    M = possible combination of transferred layers
end if
baseline: Train network A using source dataset, D_S as shown in the baseline approach.
for each model M in the ensemble of TL **do**
    transfer: transfer features from network A to new network B as shown in the transfer learning approach
    Compute posterior probabilities $P_T(y|x)$ for target dataset, D_T.
end for
{Combine all the posterior probabilities $P_T(y|x)$ of each model, M}
Compute $y = argmax \sum_{M_i \in M} P_T(y|x)$

4 Pre-processing of Datasets

We test EDTL on three different types of tasks i.e., character, object and biomedical image recognition using four original datasets[1]. To evaluate all possible TL cases as listed in Table 1, we modified the four original datasets into nine different datasets as listed in Table 2.

4.1 Character Recognition Dataset Processing

We evaluate the framework in two different settings for the character recognition task: We use the *MNIST* dataset P_L which has 60,000 training and 10,000 testing instances with labeled hand-written digits from 0 to 9. Additionally, the

[1] We would like to acknowledge researchers making available their datasets, Center for Neural Science, New York University for MNIST; Microsoft Research India for Chars74k; LISA labs, University of Montreal, Canada for BabyAI shapes; and Broad Institute of Harvard and MIT for MCF7-wt breast cancer cells.

Chars74k dataset was modified to obtain *Lowercase* dataset P_{LC} labelled lowercase letters from a-to-z and, the *Uppercase* dataset P_{UC} labelled uppercase letters from A-to-Z and *Digits* P_D dataset contains digits from 0-to-9. Both MNIST and Chars74k datasets used in our experiments have images with 28 x 28 pixels and a sample of each dataset.

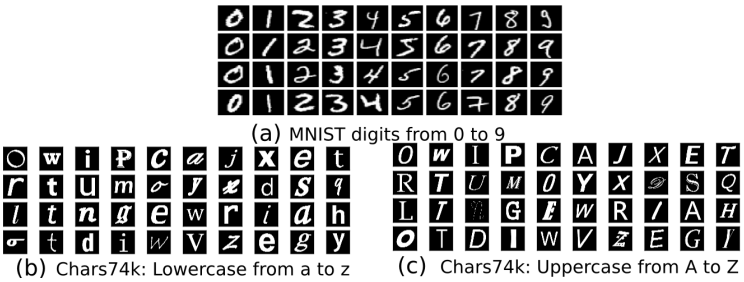


Fig. 3. Samples from character recognition tasks: (a) MNIST, (b) lowercase and, (c) uppercase

4.2 Object Recognition Dataset Processing

We generated three shapes datasets. First, the *canonical* dataset P_{Sh1} has canonical objects, i.e., equilateral triangle, circle and square. Second, the *non-canonical* dataset P_{Sh2} has non-canonical objects, i.e., triangle, ellipse and rectangle. Finally, the *curve Vs. corner* dataset P_{Sh3} has shapes with a curved surface or a corner. All three datasets used in our experiments have images with 28 x 28 pixels and a sample of each dataset.

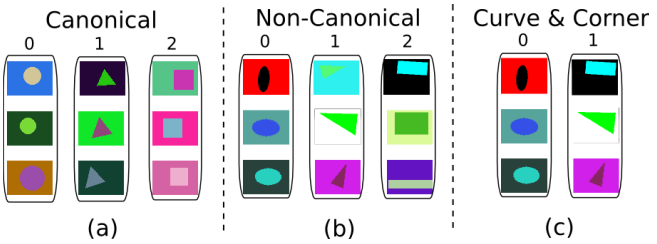


Fig. 4. Samples from various shape recognition tasks: (a) Canonical, (b) Non-Canonical and (c) Curve & corner

4.3 Biomedical Image Recognition Task

We used the BBBC021 image set [18] of genetically engineered MCF7-wt (breast cancer expressing wild-type p53) cell line. The MFC7 dataset has around 4 million cancer cells including Dimethyl sulfoxide (DMSO) control samples. The

CellProfiler software was used to extract 453 features of each of the 148,649 cells of non-control samples. Each single-cell sample was clearly labelled having one of 12 different primary mechanisms of action (MOA) from the subset of compound-concentration combinations [19]. We modified the MFC7 dataset to solve for two different tasks. The first task is to identify 12 different MOA using the single-cell features of non-control samples. The second task is to categorize 38 compounds using the same samples.

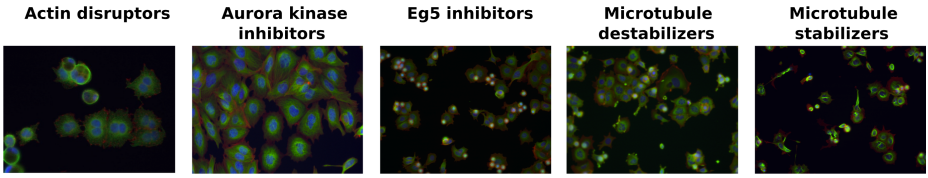


Fig. 5. Examples of the broad spectrum of heterogenic phenotypes captured of MCF7-wt cancer cell assay after compound incubation. The high-content image consists of four-wavelengths, DNA binding dye, DAPI (blue), an actin cytoskeleton marker, Phalloidin (red), tubulin antibody (green) to monitor the microtubule cytoskeleton, and the cytoplasmic marker, HCS cell mask.

Table 2. Number of instances available for each dataset

Data set		**Labels**			**Instances**		
		Ω		classes	**Train**	**Valid**	**Test**
MOA	P_{moa}	0..11	Ω_{12}	12	74,325	37,162	37,162
Compound	P_{com}	0..37	Ω_{38}	38	74,325	37,162	37,162
Lowercase	P_{LC}	a-to-z	Ω_{az}	26	13,208	6,604	6,604
Uppercase	P_{UC}	A-to-Z	Ω_{AZ}	26	13,208	6,604	6,604
Digits	P_D	0-to-9	Ω_{09}	10	13,208	6,604	6,604
MNIST	P_L	0-to-9	Ω_{09}	10	50,000	10,000	10,000
Canonical	P_{Sh1}	eqt,cir,sqr	Ω_{sh1}	3	14,000	1,000	5,000
Non-Canonical	P_{Sh2}	tri,ell,rec	Ω_{sh2}	3	14,000	1,000	5,000
Curve & corner	P_{Sh3}	rou,cor	Ω_{sh3}	2	14,000	1,000	5,000

5 Experimental Setup and Results

Training Deep Neural Network: The network we used in character recognition experiments had three hidden layers with [576, 400, 256] units in order of [bottom, middle, top] respectively, batch size of 100 and pre-training ran for a minimum of 25 epochs. The networks used in object recognition experiments also had three hidden layers with [100, 200, 300] units, batch size of 300 and pre-training ran for a minimum of 10 epochs. The networks used in biomedical image recognition experiments with [453, 906, 1359] units, batch size of 100 and pre-training ran for a minimum of 30 epochs. All the three networks have an output layer appropriate to the number of classes being considered. All hidden layers were pre-trained as denoising autoencoders via gradient descent, using

the cross-entropy cost and a learning rate of 0.001. The complete networks were fine-tuned via gradient descent, using the cross-entropy cost and a learning rate of 0.1. The fine-tuning ran until the validation error did not decrease below 0.1% or until 1000 epochs for all tasks. Our code for experiments was based on the Theano library 6 and ran with the help of an GTX 770 GPU. To determine if a result is statistically significant over ten repetition of each experiment, we used paired student t-test to calculate a p-value, which is the probability of observing an effect given that the null hypothesis is true. We marked each result in Table 3, with '*' when the result was statistically significant, i.e., if an observed p-value is lower than 0.01 (1%).

5.1 Retrain Specific DTL

In this section, we study Retrain specific DTL ($\mathbf{DTL}_r$). In this condition of DTL, we transfer all the hidden layers of the source network to the target network, i.e., transfer [1 1 1 1] and retrain only unlocked layers marked as '1', for example retrain [0 0 1 1]. We study the fragile splitting of the co-adapted neurons caused due to locking of the layer, thus stopping learning in that selected hidden layer of the target network. This avoids overfitting of the network for the target task.

Generally the features of the lower layer of the network are generic therefore they can be used to solve a broader spectrum of problem. The higher layer features are specific to the task the network was trained. We would like to re-utilize the generic features of the source network and retrain the transferred network for target specific task. In this section, we study suitable conditions such that we obtain positive transference retraining only specific layers of the target network.

We observe a consistent improvement in $\mathbf{DTL}_r$ across all the cases of transfer learning for the condition: transfer = [1111] & retrain = [1111]. We conclude that this is due to two main reasons: 1) the transferred layer weights are better than random initialization and, 2) retraining the network target task improves the chances of better generalization.

We observe statistically significant result for all conditions of $\mathbf{DTL}_r$ except for transfer = [1111] & retrain = [0001]. This still offer good generalization than the random accuracy, but is lower than in other conditions.

Ensemble of 4 $\mathbf{DTL}_r$ models gives retrain specific EDTL ($\mathbf{EDTL}_r$). We observe better average accuracy than BL and $\mathbf{DTL}_r$ conditions and results are shown in Table 3. We perform paired student t-test comparing the accuracy results $\mathbf{EDTL}_r$ with accuracy results of $\mathbf{DTL}_r$.

5.2 Transfer Specific DTL

In this section, we study Transfer specific DTL ($\mathbf{DTL}_t$). In this condition of DTL, we transfer only specific layers of the source network to the target network, for example transfer [0 0 1 1] and retrain all the layers, i.e., retrain [1 1 1 1]. We study the generic versus specific feature transference due to transferring of the

Table 3. Percent average classification accuracy obtained for all three possible transfer learning cases; 6 different experiments are performed on three different types of tasks i.e., character, object and biomedical image recognition; We compare established frameworks i.e., Baseline (BL), retrain specific DTL ($\mathbf{DTL}_r$), and transfer specific DTL ($\mathbf{DTL}_t$) with our approach, retrain specific EDTL ($\mathbf{EDTL}_r$), transfer specific EDTL ($\mathbf{EDTL}_t$), and Ensemble of DTL (EDTL); the difference between two datasets distribution and is given by Jensen-Shannon divergence (JSD)

	Marginal	$P_S(X) \neq P_T(X)$		$P_S(X) = P_T(X)$		$P_S(X) \neq P_T(X)$	
	Labels	$Y_S = Y_T$		$Y_S \neq Y_T$		$Y_S \neq Y_T$	
	TL case	I		II		III	
	Experiment	①	②	③	④	⑤	⑥
	Source	Non-Canonical	MNIST	Non-Canonical	COMP	MNIST	MNIST
	Target	Canonical	Digit	Curve & corner	MOA	Lower	Upper
	JSD	0.99	0.99	0	0	0.80	0.79
Approaches		**Avg Acc**	**Avg Acc**	**Avg Acc**	**Avg Acc**	**Avg Acc**	**Avg Acc**
BL		99.49(0.32)	97.74(0.09)	98.35(0.27)	96.38(0.5)	94.34(0.13)	94.93(0.13)
Retrain Specific DTL							
	transfer retrain						
$\mathbf{DTL}_r$	[1111] [1111]	99.51(0.17)	97.92(0.27) *	99.00(0.46)	97.54(0.40) *	94.71(0.22) *	94.94(0.23) *
	[1111] [0111]	96.92(1.72) *	98.06(0.17) *	97.06(1.47) *	98.07(0.18)	94.52(0.24) *	94.74(0.15) *
	[1111] [0011]	96.60(1.64) *	98.14(0.18) *	96.79(1.53) *	98.23(0.17)	93.92(0.29) *	94.10(0.15) *
	[1111] [0001]	95.78(1.91) *	97.36(0.51) *	96.46(1.68) *	98.16(0.22)	89.36(0.83) *	89.57(0.99) *
$\mathbf{EDTL}_r$		**99.57**(0.13)	**98.62**(0.14)	**99.27**(0.31)	**98.24**(0.27)	**95.11**(0.18)	**95.38**(0.18)
Transfer Specific DTL							
	transfer retrain						
$\mathbf{DTL}_t$	[1111] [1111]	99.51(0.17) *	97.93(0.27) *	99.00(0.46) *	97.54(0.40)	94.71(0.22)	94.94(0.23) *
	[0111] [1111]	99.73(0.12) *	97.09(0.34) *	**99.60**(0.16) *	96.76(1.22)	92.80(0.26) *	93.96(0.15) *
	[0011] [1111]	86.83(17.26)	97.29(0.30) *	68.81(5.71)	96.71(0.46) *	93.28(0.28) *	93.97(0.33) *
	[0001] [1111]	99.84(0.08)	97.46(0.17) *	98.86(0.38) *	96.81(0.65)	93.40(0.17) *	94.38(0.16) *
$\mathbf{EDTL}_t$		**99.91**(0.03)	**98.18**(0.16)	99.58(0.16)	**97.54**(0.53)	**94.56**(0.11)	**95.26**(0.10)
EDTL		99.86(0.03)	**98.87**(0.10)	99.52(0.16)	98.12(0.31)	**95.18**(0.06)	**95.70**(0.16)

layer, thus reusing the features for the target task. This not only speeds up the training but also improves the accuracy of the network.

We observe that $\mathbf{DTL}_t$, even for condition when only the logistic regression layer is transferred and retaining the whole target network with backpropagation algorithm had better accuracy than the BL as shown in Table 3.

Ensemble of 4 $\mathbf{DTL}_t$ models gives transfer specific EDTL ($\mathbf{EDTL}_t$). We observe better average accuracy than BL and $\mathbf{DTL}_t$ conditions. Results are shown in Table 3.

5.3 Ensemble of both $\mathbf{EDTL}_r$ and $\mathbf{EDTL}_t$

We observe significant improvements in average accuracy using EDTL over both $\mathbf{EDTL}_r$ and $\mathbf{EDTL}_t$ using all the conditions as listed in Table 3 except for the transfer learning case II. Firstly, we observe that in $\mathbf{EDTL}_r$, 6 out of 6 experiments obtains better accuracy than BL and other established DTL approaches. Secondly, we observe that in $\mathbf{EDTL}_t$, 5 out of 6 experiments obtains better accuracy than BL and other established DTL approaches. Finally, we observe in EDTL, 3 out of 6 experiments obtains better accuracy than BL and other established DTL approaches.

6 Conclusions and Discussion

We propose an ensemble of transfer learning approaches using 9 datasets with varied image recognition tasks like character, object and biomedical image recognition. We make several contributions as listed below:

1. We analyse all possible cases of transfer learning, i.e., based on change in distribution and based on change in classification task between the source and the target domains.

2. The experimental analysis of the retrain specific DTL approaches across all possible cases of transfer learning show that the condition: transfer all layers and retain all layers, obtains better overall accuracy not only than baseline but also compared to other DTL_r conditions. This is due to two main reasons: 1) the transferred layer weights are better than random initialization and, 2) retraining the network target task improves the chances of better generalization, i.e., by forcing splitting of fragile co-adapted neurons the network avoids overfitting to the target task.

3. We observe that transfer specific DTL approaches obtain better overall accuracy than baseline but not as good as retrain specific DTL, as fine-tuning of randomly initialized weights forces the solution to local minima.

4. The experimental analysis of retrain specific EDTL, transfer specific EDTL and EDTL approaches show that $EDTL_r$, ensemble of posterior probabilities of four DTL_r models, obtain a statistically significant better accuracy than individual DTL_r. EDTL outperforms baseline and other DTL approaches both when the distributions and task are different.

In future we would like to explore the possibility of training multiple source domain problems and combine them under ensemble of deep transfer learning framework.

Acknowledgments. This work was financed by FEDER funds through the *Programa Operacional Factores de Competitividade* COMPETE and by Portuguese funds through FCT Fundação para a Ciência e a Tecnologia in the framework of the project PTDC/EIA-EIA/119004/2010.

References

1. Thrun, S.: Learning to learn: Introduction. In Learning To Learn (1996)
2. Caruana, R.: Multitask learning. Machine Learning **28**(1), 41–75 (1997)
3. Daumé III, H., Marcu, D.: Domain Adaptation for Statistical Classifiers. J. Artif. Intell. Res. (JAIR) **26**, 101–126 (2006)
4. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: Proc. of the ACM Conference on (ICML), pp. 759–766 (2007)
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. proceedings of the IEEE **86**(11), 2278–2324 (1998)
6. Hinton, G.E., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. The Journal of Neural computation **7**, 1527–1554 (2006)
7. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
8. Bengio, Y., et al.: Towards Biologically Plausible Deep Learning. arXiv preprint arXiv:1502.04156 (2015)
9. Kandaswamy, C., Silva, L., Alexandre, L., Sousa, R., Santos, J.M., Marques de Sá, J.: Improving transfer learning accuracy by reusing Stacked Denoising Autoencoders. In: IEEE Conference on Systems Man and Cybernetics. IEEE (2014)
10. Kandaswamy, C., Silva, L.M., Alexandre, L.A., Santos, J.M., de Sá, J.M.: Improving deep neural network performance by reusing features trained with transductive transference. In: Wermter, S., Weber, C., Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., Villa, A.E.P. (eds.) ICANN 2014. LNCS, vol. 8681, pp. 265–272. Springer, Heidelberg (2014)
11. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
12. Kandaswamy, C., Silva, L., Cardoso, J.S.: Source-target-source classification using Stacked Denoising Autoencoders. In: Proc. of the 7th Iberian Conference on Pattern Recognition and Image Analysis, Santiago de Compostela, Spain, June 2015
13. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 512–519. IEEE (2014)
14. Deng, L., Platt, J.C.: Ensemble deep learning for speech recognition. In: Proceedings of the Annual Conference of International Speech Communication Association (INTERSPEECH) (2014)
15. Abdullah, A., Veltkamp, R.C., Wiering, M.A.: An ensemble of deep support vector machines for image categorization. In: International Conference of Soft Computing and Pattern Recognition, SOCPAR 2009, pp. 301–306. IEEE (2009)

16. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2012)
17. Lin, J.: Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory **37**, 145–151 (1991)
18. Ljosa, V.: Katherine L. Sokolnicki, and Anne E. Carpenter.: Annotated high-throughput microscopy image sets for validation. Nat Methods **9**(7), 637 (2012)
19. Ljosa, V., et al.: Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. Journal of biomolecular screening (2013)
20. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley Press (2004)

Development of a Power Output Forecasting Tool for Wind Farms Based in Principal Components and Artificial Neural Networks

P. del Saz-Orozco[1(✉)], J. Fernández de Cañete[1], and R. Alba[2]

[1] System Engineering and Automation Department, University of Malaga,
Edificio de Ingenierías C/Dr. Ortiz Ramos, s/n Campus de Teatinos s/n, 29071 Malaga, Spain
pablo.soh@gmail.com
[2] Gas Natural Fenosa, Av. de San Luis, 77, Madrid, Spain

Abstract. The main objective of the study here presented consists in developing a mathematical forecasting model of the available wind power output for an eight-hour horizon in wind farms that may be affected by inclement meteorological environments where the surface of the wind turbine blades can suffer of ice accumulation. These events may depend on several factors as air temperature, relative humidity, barometric pressure or wind speed, among others. In this way a precise model depending on the referred variables will allow predicting with higher accuracy the available power at the plant when the referred events may occur. A model based in neural networks for the prediction of the available power output of an experimental wind farm has been developed and tested using real data. The proposed model outperforms other professional commercial models.

Keywords: Wind power forecasting · Artificial Neural Networks · Principal components analysis

1 Introduction

Nowadays nobody doubts about the importance of Wind Power Generation within the whole electric power sector due to several reasons as, among others, being plentiful renewable once installed as it does not emit during the generation process any greenhouse gas and can be widely distributed as it does not use big areas of land per MW installed.

As a consequence the cumulative global wind power installed capacity has grown by around twenty times in the last 14 years [1]. On the other hand, these systems also introduce additional uncertainty in the system as the wind speed and direction forecast is not accurate at a 100%. This is a very important issue due to the specific characteristics of the electric power industry in which the generation should be always in equilibrium with the consumption [2].

Wind power output forecast is intimately related with wind speed forecast as the wind speed can be converted into power through the power curve of the turbines.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 349–360, 2015.
DOI: 10.1007/978-3-319-19258-1_30

There are several strategies in the forecasting of power output in Wind Power Farms depending on the time scale and the approach for their development [3]. The modeling methods can be divided into two main categories, the physical methods and the statistical methods [4]. The first kind of methods are based in a wind speed forecast taking into account physical variables as, for instance, the terrain or the temperature, among others. The second type of methods uses statistical approaches to forecast mean power production using historical time series of wind data. The first kind of methods are used normally for short term horizons (mean daily or hourly wind forecasts) while the second group is more suited for longer time horizons.

Among the first kind of systems there are several approaches as using computational fluid dynamics approaches [5], HIRLAM models [6] or persistence models [7]. And there are several approaches within statistical models including autoregressive models [8], moving average models [9], autoregressive moving average models [10], autoregressive integrated moving average models [11] or Kalman filters [12].

In the last years different artificial intelligence techniques have also been applied to wind power forecasting due to their recent developments. Artificial intelligence methods involve, mainly, the use of artificial neural networks [13], fuzzy logic and their associated hybrids schemes as neuro-fuzzy networks [14], Random Forest [15], support machine regressors [16], or hybrid schemes [17].

Some additional specific issues arise when the wind farm is settled in areas where the climate is inclement for a substantial part of the year. Icing problems may occur when the blades of the turbines are exposed to atmospheric conditions in which liquid water content and near freezing temperatures and/or high convective cooling rates exist. These conditions depend on several factors as air temperature, relative humidity or wind speed, among others. In this kind of cases the available wind power will not only depend in the wind speed and direction but also in other variables that will not allow the system to work at full load [18]. Therefore a precise model of the conditions of ice formation in the blades depending on the referred variables will allow predicting with higher accuracy the available power at the plant at each hour, impacting heavily in the strategic bidding and operational maintenance of the plants in some months in which the wind usually blows at a higher speed.

In this paper we describe the development of a power output forecasting tool based on artificial neural networks that has been trained in order to predict the available power output with the possibility of ice formation on the blades from meteorological forecasting values for a horizon of up to 8 hours.

The document is divided in the following sections. In Section II the Iberian electricity market will be briefly described in order to understand the final time horizon needed for the forecasting tool that is proposed. In following section the general structure of the forecasting tools and mathematical tools that have been used in its development will be presented. Then the main obtained results are presented in the next section and main conclusions are presented also defining new future guidelines lines in which this study may be continued.

2 The Iberian Power Market

The Iberian electricity market consists mainly of a sequence of markets with different time horizons. The daily market is the main market for purchasing electricity in the Iberian System. Each day around 90% of the energy that is going to be delivered in the following day is traded in this market [19]. On the other hand, the intraday markets are adjustment markets. In this kind of markets both energy buyers and sellers may reconfigure their previous positions according to the new scenarios and power forecast developed during the day. The intraday Market is structured into six different sessions. Its scheme is shown in figure 1.

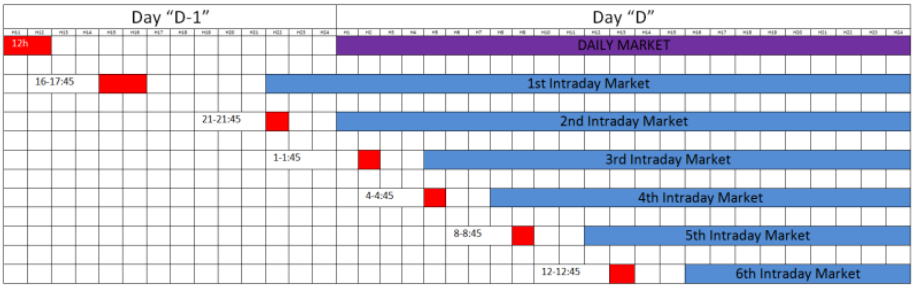


Fig. 1. Intraday market schedule

So, it can concluded that the first intraday market will be the last chance to change the position of the energy consumed and/or produced from 22 h. and 24 hours and the strategic bidding should be finished at 17:45. In this way, if a seller/buyer wants to change his position with a new forecast it should practically cover at least 7 hours (in order to prepare the offer and send it among other activities involved in the bidding process). Therefore a proper forecast tool for the second intraday market should also cover 7 hours. We can see a summary of the horizon requirements for each intraday market in table 1.

The electricity due to its nature of not usually being possible to be stored in large amounts has to meet properly the generation and the demand in real time. So, all the sources of imbalances between energy production and consumption have to be properly corrected in order to assure the stability of the system.

The steady growth of the share of wind power generation presents the main drawback of presenting significant variation on the production on short term time scales and the uncertainties in the forecast of the wind production. The balancing costs depend on the prediction errors. Therefore the growing share of intermittent generation sources with high uncertainties increase the demand for reserve and response operation in a way these costs could increase dramatically [20]. Hence, a reduction in the prediction errors can impact heavily in the economic performance of wind farms.

Table 1. Time Horizon Requirements according to the intraday markets it is aimed

Intraday Market	Horizon (h.)	Intraday Market	Horizon (h.)
1st	7	4th	7
2nd	7	5th	7
3rd	6	6th	8

3 Data Description and Mathematical Tools

In this section the different internals of the forecasting tool will be described. Firstly the origin of the data that have been used for training and validating the forecasting tool will be depicted and afterwards the basics of the different mathematical tools that have been used will be defined in a way the capabilities and possible drawbacks of the final system could be determined.

3.1 Data Origin

The developed model presents two different types of data sources. On the first hand the model uses weather forecasting data in order to infer the predicted output from the forecasted weather variables. On the other hand the collected data from the wind power plant's PLCs will be used both for training purposes and for testing the model output. In the next paragraphs the characteristics of both data-sources will be briefly described.

Weather Forecasting Data

In the case here presented, the studied wind farm is located in the region of Leon sited in the autonomous community of Castile and Leon in the northwestern part of Spain. This specific region is characterized for presenting high wind speeds and where occasional icing events occur [21]. Specifically, the location is shown as A in figure 2.

In order to develop the training and validation sets the data from a National Weather Forecast models for the period from 9:00 of 4th April 2013 to 10:00 6th May 2013 were stored. This period of time was selected due to the relatively high probability of ice events and high variability of wind speed. The referred model provides the forecasted meteorological values for the following 44 hours for the following variables: temperature, wind direction, mean wind speed, wind gust, mm of rain, the percentage of occurrence of clouds, the relative humidity and the atmospheric pressure. And the stored data correspond to the location where the wind farm is located and from five surrounding locations.

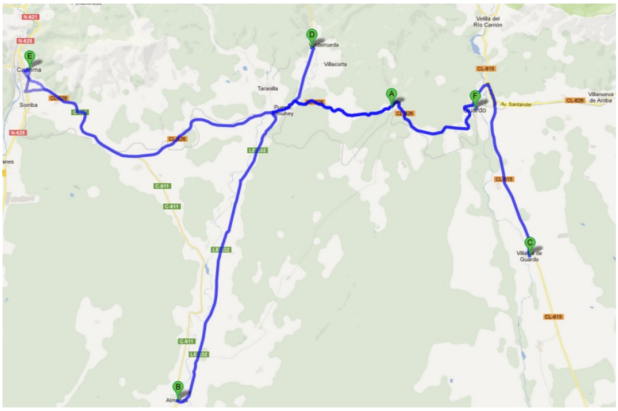


Fig. 2. Situation of the different villages used in the forecasting tool

Table 2. Distances from the different locations to the location where the wind farm is sited

Location	Distance (km)	Location	Distance (km)
B	22	E	23
C	17	F	7
D	9		

SCADA Data Description

The data used in this research were generated at a wind farm with nine turbines. The data was collected by a SCADA system installed at each wind turbine. Each SCADA system collects 381 variables per tower. The variables are the input to the system, alarms that inform of malfunctions in the system, state variables and output values as the generated power. This data is sampled at high frequency (higher than 1 Hz) and afterwards it can be mathematically processed. In the specific case presented in this paper the hourly power output in order to train and validate the forecasting tool and also the values of the alarm that informs about ice events were stored.

3.2 Mathematical Tools

As it has been stated in previous sections the available wind power forecast in a given farm is a highly non-linear dynamic system based on meteorological predictions as, among others, the wind speed or the wind direction and other variables as, for instance, the performance of the generator or the blade position. In the present study we are also considering the available power reduction due to the presence of ice in the blades, so we should also consider other atmospheric variables.

Artificial Neural Networks are a mathematical approximation on how a biological brain is supposed to work. The human brain can be seen as a truly remarkable parallel computer, able to process incomplete information and generalize conclusions at an

incredibly rapid rate [22]. The use of artificial neural networks is motivated by their learning capability and their ability to model dynamic systems using a black-box approach from Input-Output data.

On the other hand, the main objective that is under Principal component analysis [23] is to reduce the dimensionality of a set of data if those data come from variables that are interrelated. Obviously the reduction is done retaining as much information present in the original dataset. This is achieved by transforming the original set to a new set of variables (the principal components) ordered in a way that the first few retain most of the variation present in the original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it should be orthogonal to the preceding components. Another very interesting feature of the Principal Component Analysis is that it de-correlates the original data matrix.

3.3 General Structure of the Forecasting Tool

In this section the main characteristics of the neural network have been developed for forecasting the available wind power taking into account other weather variables in addition to the wind speed and direction will be presented. Firstly the followed iterative process in order to select the input variables to each network will be presented. Then the structure of the networks and the training process will be defined.

In the figure 3 the main blocks of the whole forecasting tool for an n-hour time horizon can be seen. In the block sited at the left the data projection and selection is represented and at the right end the block that represents the forecasting neural network can be seen. In the figure P_t is the power measured at time t, $\tilde{P}_{t+i}$ are the predicted production for the hour t + 1, I_j is the input j to the neural input and $\tilde{V}_{t+k}$ are the meteorological predictions for the hour t + k for all the geographical sited as explained in a previous section.

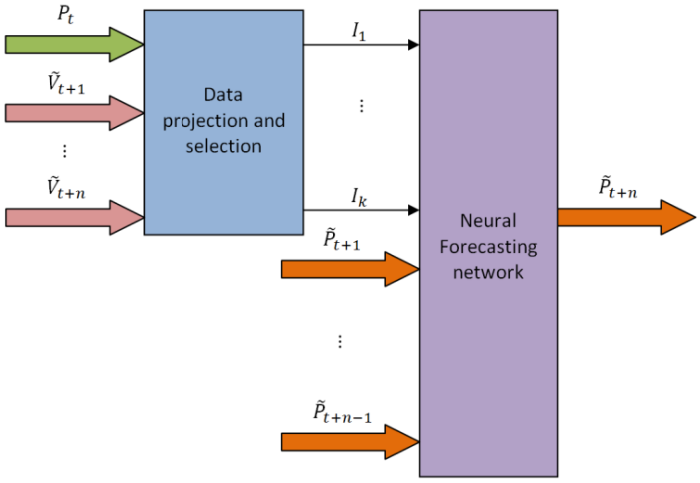


Fig. 3. Structure of the forecasting tool for a given time horizon

Data Selection

In the present study the dimensionality of the input space will be reduced using the principal components of the forecasted values of the different meteorological variables. The set that accumulates 95% of the overall information will be preliminarily selected which is the task of the block "Data projection and selection" in figure 3.

Table 3. Input vector dimension per time horizon's Neural Network

Time Horizon	Number of Variables	Time Horizon	Number of Variables
1	8	5	14
2	9	6	15
3	9	7	18
4	12	8	20

With the referred set a neural network will be trained and the errors will be evaluated. Afterwards in an iterative process those variables that present an absolute value of their correlation with the error higher than 0.1 will be also selected. In this way in order to select all the variables we will proceed in a recursive way until the highest absolute value of any correlation between the non-selected variables with the output error is lower than 0.1.

Finally in the table 3 we can see the final dimension of the input set to each one of the different neural networks.

Structure of the Different Networks and Training Process

Once the input set for every network has been selected, the next steps consist in defining the neural network structure and afterwards train the network. In the presented case a feed forward network with a single hidden layer network has been selected for all the networks.

Finding the optimal dimension is a crucial task in order to achieve a good identification model. A network that does not have enough parameters may be unable to learn the required task, while choosing a network that is larger than necessary may lead to an over parameterized network. A trial and error process in order to determine the size of the hidden layer of each network has been chosen. In table 4 the number of hidden nodes in each network having all of them a logistic activation function has been listed.

Table 4. Number of hidden neurons per network

Time Horizon Network	Number of hidden Neurons	Time Horizon Network	Number of hidden Neurons
1	7	5	5
2	7	6	5
3	6	7	5
4	10	8	20

4 Results

In this section the obtained results in the forecasting of the available wind production for the collected experimental data will be presented and analyzed. In order to validate the proposed tool the neural-based system's output against the actual stored output of the wind farm will be tested, using both the mean error and the correlation between the actual output and the output predicted by the model.

In order to justify the need of using the referred eight atmospheric variables we are going to compare the results of two different forecasting models. The first one will use as input all the available variables and afterwards projecting them and selecting the principal components variables as it was explained in a previous section. The second one will use as inputs the forecast of the mean wind speed, the wind gust and the wind direction. For each time horizon, the network structure (number of hidden layers and neurons per layer) is the same.

For both models the error function for a given hour will be defined as the absolute value of the difference between the actual production and the output of the model as follows:

$$\bar{E} = \frac{1}{n}\sum(|O_i - P_i|) \tag{1}$$

Where O_i is the model and P_i is the actual production and n is the number of samples (taking into account that the maximum production is 18000 KWh). Additionally, the correlation between the actual output of the wind farm and the output predicted by the neural based model will also be used and it is defined as:

$$R = \frac{n\sum O_i P_i - (\sum O_i)(\sum P_i)}{\sqrt{(n\sum {O_i}^2 - (\sum O_i)^2)(n\sum {P_i}^2 - (\sum P_i)^2)}} \tag{2}$$

A neural network per horizon hour has to be trained and tested and the final model will be the aggregated model considering each hour depending to their distance to the nearest previous intraday market, as it was defined in table 2.

In all the cases the training methodology will be a Levenberg-Marquardt training phase [22] with 1000 iterations. The training set will consist in 60% of the whole set of valid samples while the validation set will consist in the remaining 40% of the

whole set. The training set samples for each time horizon have been selected randomly from the complete set.

The results for the training and validation phases are shown in table 5. The model using all the meteorological variables is referred as model 1 while the model that uses only the wind speed, the wind direction and the wind gust as inputs is referred as model 2.

Table 5. Mean error for the validation phase

	Validation phase. Model 1.		Validation phase. Model 2.	
Time Horizon	Mean error (kWh)	Correlation	Mean error (kWh)	Correlation
4	852,48	0,9325	1454,68	0,8180
5	929,51	0,9269	1468,24	0,8039
6	1042,00	0,9083	1593,08	0,7952
7	1210,22	0,8865	1767,82	0,7561
8	1261,38	0,8760	1895,98	0,7165

As it was expected for both cases the mean error at the validation phase increases as it also does the time horizon. The maximum mean absolute error for model 1 is lower than the minimum mean absolute error for model 2 while the minimum correlation for model 1 is higher than the maximum correlation for model 2. So it can be concluded that in this case the usage of more variables in the forecasting model is justified.

For the model 1, the maximum mean error occurs for a time horizon of 8 hours with a value of 1261,38 kWh (which represents a 7% of the maximum possible output of the wind farm). The minimum correlation coefficient also occurs for the 8-hour time horizon with a value of 0,876. In the figure 5, the forecasted value for a time horizon of 8 hours is shown.

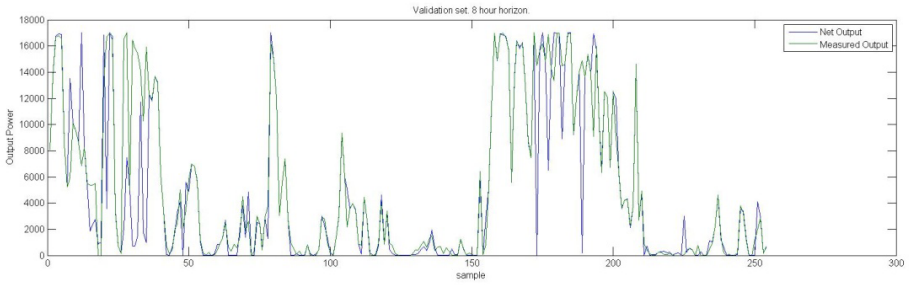


Fig. 4. Validation results for 8-hour horizon

Once the networks for the different time horizon have been trained and validated, the next natural step will consist in aggregate the different networks in a single model taking into consideration in which hour its forecasting time horizon depending on the intraday market clearing hours as it was shown in table 2. In this way the resulting system will consist in aggregating the previously presented ANN in a single predicting model. Applying the referred approach we obtain the results shown in figure 5.

In the upper figure 5 the comparison between the forecasted output (in blue) and what the farm really produced in those hours (in green) can be seen. Also the production in those hours in which the farm was limited (in red) due to an excess of hydro power that occurred in those periods of time in the Iberian overall power system is shown. Obviously, those hours were not used in the training set or in the validation set and were not included in the error evaluation listed in table 5. In the lower part of figure 5 the histogram of the error values is shown.

The system is capable of predicting a low output when the farm is producing less due to meteorological weather conditions (hours 370-410, 455-500 and 670-768 in which ice and snow episodes occurred and the ice alarm of at least one of the towers was on). The lower part of figure 5 shows the histogram of the error function for the period of time that has been studied. Finally the overall system presents a mean error of $1.077 \cdot 10^3 kWh$ that is more than a 47% lower than the mean error the company obtained for this plant in the actual bidding process for the same given period of time ($2.0302 \cdot 10^3 kWh$). In this way it can be concluded that the presented strategy looks as a promising approach for predicting the wind power output of real wind farms with a relatively long horizon for cases in which the output may depend heavily not only in the wind speed and direction but also in other meteorological variables.

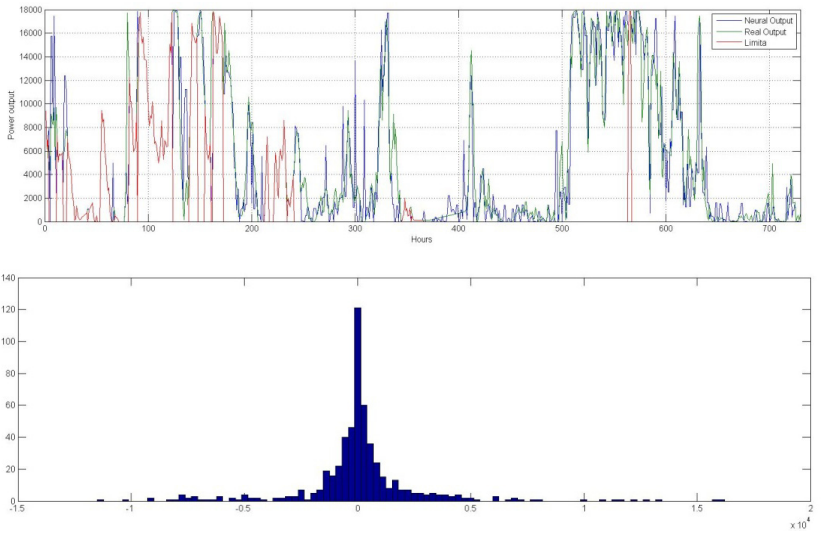


Fig. 5. (Upper) Forecasting tool output. (Lower) Output Error Histogram.

5 Conclusions and Future Work

In the study presented in this paper the development a mathematical model that takes into account more weather variables than just the wind speed and direction in order to predict the available power in harsh weather situations is presented. Therefore this information can be included in the intraday bidding process so the day-ahead strategy can be changed with more accurate values of the forecasted power output.

The model is based in neural networks due to their capability of approximating any functional relationship from incomplete or noisy data sets. Principal components analysis technique has been used in order to preprocess the data. The use of neural networks in this specific field is not new as it was presented in a previous section, but this model was developed with a longer time horizon than the usual systems that appear in the scientific literature in order to adapt the resulting system to the Iberian intra-day market characteristics and uses more variables in order to consider the possible formation of ice in the blades.

As it has been shown in the results section, ANN-based techniques seem to be very powerful approaches in the forecasting of wind power output even with time horizons fully compatible with real electricity markets.

The future works are directed in the usage in real time of the forecasted values in order to follow an optimal bidding strategy.

If a forecasting model of the market prices is available forecasting both prices and imbalance costs, the bidding strategy can be seen as an optimization problem given a position in the daily market [24]. Therefore an interesting future work of the present study could be the development of a complete market model including the wind forecast in order to optimize the bidding strategy.

References

1. Gasch, R., Twele, J. (ed.): Wind Power Plants. Fundamentals, Design, Construction and Operation, 2nd edn. Springer-Verlag (2012)
2. de Sistiernes, F.: Quantifying the Combined impact of Wind and Solar Power Penetration in the Optimal Generation Mix and Thermal Power Plant Cycling. Young Energy economists and Engineers (2011)
3. Giebel, G., Brownsword, R., Kariniotakis, G., Denhard, M., Draxl, C.: The State-of-the-art in Short-Term prediction of Wind Power. In: Deliverable D-1.2 Anemos.plus project, 6th Framework Program (2011)
4. Foley, A., Leahy, P., Marvuglia, A., McKeogh, E.: Current methods and advances in forecasting of wind power generation. Renewable Energy **37**, 1–8 (2012)
5. Magnusson, M., Wern L.: Wind energy predictions using CFD and HIRLAM forecast. In: Proceedings of the European wind energy conference EWEC2001. Copenhagen, Denmark (2001)
6. Langberg, L.: Short term prediction of power production from wind Farms. Journal of Wind Engineering and Industrial Aerodynamics **80**, 207–220 (1999)
7. Langberg, L.: Short term prediction of local wind conditions. Journal of Wind Engineering and Industrial Aerodynamics **89**(3/4), 235–245 (2001)

8. Pinson, P., Madsen, H.: Adaptive modeling and forecasting of offshore wind power fluctuation with Markov-switching autoregressive models. Journal of forecasting **31**(4), 281–313 (2012)
9. Karki, R., Hu, P., Billinton, R.: A Simplified wind power generation model for reliability Evaluation. IEEE transaction on Energy conversion **21**(2), 533–540 (2006)
10. Erdem, E., Shi, J.: ARIMA based approaches for forecasting the tuple of wind speed and direction. Applied Energy **88**(4), 1405–1414 (2011)
11. Chen, P., Pedersen, T., Bak-Jensen, B., Chen, Z.: ARIMA-based time series model of stochastic wind power generation. IEEE transactions on power systems **25**(2), 667–676 (2010)
12. Cassola, F., Burlando, M.: Wind Speed and wind energy forecast through Kalman filtering of Numerical Weather prediction model output. Applied Energy **99**, 154–166 (2012)
13. Methaprayoon, K., Yingvivatanapong, C., Lee, W., Liao, J.: An Integration of ANN Wind Power Estimation Into Unit Commitment Considering the Forecasting Uncertainty. IEEE Transactions on Industry Applications **43**(6), 1441–1448 (2007)
14. Pinson, P., Siebert, N., Kariniotakis, G.: Forecasting of regional wind generation by a dynamic fuzzy-neural networks based upscaling approach. In: European Wind Energy Conference and Exhibition, EWEC 2003 (2003)
15. Fugon, L. Juban, J.; Kariniotakis, G.: Data Mining for wind power forecasting. In: European Wind Energy confenernce and Exhibition, EWEC 2008 (2008)
16. Kusiak, A., Li, W.: Short-term prediction of wind power with a clustering approach. Renewable Energy **35**, 2362–2369 (2010)
17. Rahmani, R., Yusof, R., Seyedmahmoudian, M., Mekhilef, S.: Hybrid technique of ant colony and particle swarm optimization for short term wind energy forecasting. Journal of Wind Engineering and Industrial Aerodynamics. **123**(part A), 163–170 (2013)
18. Homola, M.: Impacts and causes of icing on wind turbines. In: Interreg III project Wind Energy in the BSR: Impacts and Causes of Icing on Wind Turbines, European Union Project. Narvik University College (2005)
19. Febrel, A.: Estudio de Prestación de Servicios Complementarios con parques Eólicos. Bachelor Thesis (Industrial Engineering), Comillas Pontifical University, Madrid (2006)
20. Borggrefe, F., Neuhoff, K.: Balancing and Intraday Market Design: Options for Wind Integration. Climate Policy Initiative working paper (2011)
21. Tammeling, B., Cavaliere, M., Holttinen, H., Morgan, G., Seifer, H., Säntti, K.: Publishable report. Wind Energy production in cold Climate. Framework of the Non-Nuclear Energy programme (1998)
22. Jang, R., Sung, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A computational Approach to Learning and Machine Intelligence. Prentice Hall (1997)
23. Jolliffe, I.T.: Principal Component Analysis 2nd edn. Springer Series in Statistics (2002)
24. Zugno, M., Jónsson, T., Pinson, P.: Trading wind energy on the basis of probabilistic forecasts both of wind generation and of market quantities. Wind Energy **16**(6), 909–926 (2013)

CO^{2}RBFN-CS: First Approach Introducing Cost-Sensitivity in the Cooperative-Competitive RBFN Design

María Dolores Pérez-Godoy[1(✉)], Antonio Jesús Rivera[1], Francisco Charte[2], and Maria Jose del Jesus[1]

[1] Department of Computer Science, University of Jaén, Jaén, Spain
{lperez,arivera,mjjesus}@ujaen.es
[2] Department of Computer Science and Artificial Inteligence, University of Granada, Granada, Spain
francisco@fcharte.com

Abstract. The interest in dealing with imbalanced datasets has grown in the last years, since they represent many real world scenarios. Different methods that address imbalance problems can be classified into three categories: data sampling, algorithmic modification and cost-sensitive learning. The fundamentals of the last methodology is to assign higher costs to wrong classification classes with the aim of reducing higher cost errors.

In this paper, CO^{2}RBFN-CS, a cooperative-competitive Radial Basis Function Network algorithm that implements cost-sensitivity is presented. Specifically, a cost parameter is introduced in the training stage of the algorithm. This parameter modifies the learning rate of the weights taking into account the right (or wrong) classification of the instance, the type of class (majority or minority) and the imbalance ratio of the data set.

Keywords: RBFNs · Imbalanced data sets · Cost sensitive

1 Introduction

Nowadays, there exist many real applications represented by data sets where the frequency (number of instances) of certain classes substantially exceeds the frequency of the remaining classes. Furthermore, in imbalanced data sets [7], as they are typically known, the importance resides in the fact that a minority class usually represents the concept of interest, for example the intruder in an intrusion detection system; whereas the other class represents the counterpart of that concept (standard users).

Standard classifiers often show weaknesses when addressing the imbalance problem, having a bias towards the majority classes. This is due to the mechanisms inside these classifiers that benefit the right classification of the majority classes to achieve a better accuracy metric. The methods proposed for dealing with the imbalance problem can be categorized into three groups [19]: data

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 361–373, 2015.
DOI: 10.1007/978-3-319-19258-1_31

sampling, algorithmic modification and cost-sensitive learning. The last approach considers higher costs for the misclassification of examples of the majority classes trying to minimize higher cost errors.

Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network (ANN) paradigms in the machine learning field. An RBFN is a feed-forward ANN with a single layer of hidden units, called radial basis functions (RBFs) [5]. The overall efficiency of RBFNs has been proved in many areas [6] such as as classification [25,26], regression [2,8] or time series forecasting [10,22].

Authors have developed an algorithm for the cooperative-competitive design of Radial Basis Functions Networks, CO^{2}RBFN [25], that has been successfully used in classification tasks. The base version of CO^{2}RBFN has also obtained outstanding results in imbalanced problems [23]. Cost-sensitivity is one of the main methodologies addressing the imbalance problem. In this paper CO^{2}RBFN-CS, a first cost-sensitive version of CO^{2}RBFN, is presented. CO^{2}RBFN-CS is based on introducing a cost in the training stage of CO^{2}RBFN, modifying the learning rate of the weights. This costs depends on the right (or wrong) classification of the instance, the type of class (majority or minority) and the imbalance ratio of the data set. The experimentation has been performed in two stages, firstly CO^{2}RBFN-CS is compared with its base version, CO^{2}RBFN, and secondly, CO^{2}RBFN-CS is compared with existing cost-sensitive methods referenced in the bibliography.

The text is organized as follows. In Section 2, the imbalanced data sets environment is described. Section 3 introduces the cost sensitive learning. CO^{2}RBFN, is described in Section 4 and the new method proposal, CO^{2}RBFN-CS, is explained in Section 5. Finally the analysis of the experiments and the conclusions are shown in Sections 6 and 7, respectively.

2 Imbalanced Data Sets

In classification tasks, we are dealing with imbalanced data sets when the data do not have an equitable distribution among the different classes of the problem. Particularly, with a data set of only two classes, the imbalance problem occurs when one class is represented by a large number of examples, while the other is represented by only a few [7].

Classification in an imbalanced data sets environment is a difficult and important task. It is difficult, due to standard classifier algorithms having a bias towards the majority (negative) class and usually the minority (positive) class represents the concept of interest. On the other hand, many real applications present an imbalanced data sets scenario.

In the specialized literature, imbalanced data sets are managed as a whole or are characterized according to their degree of imbalance using the imbalance ratio (IR), which is defined as the ratio of the number of instances of the majority class and the minority class.

Imbalanced data sets can be categorized taken into account the IR level [11]: data sets with a low imbalance when the instances of the positive class are

between 10 and 40% of the total instances (IR lower than 9) and data sets with a high imbalance where there are no more than 10% of positive instances in the whole data set compared to the negative ones (IR higher than 9).

In order to deal with this problem, some approaches have been proposed. These approaches can be categorized into three major groups [19]:

- Data sampling: In which the training instances are modified in such a way to produce a more or less balanced class distribution that allow classifiers to perform in a similar manner to standard classification.
- Algorithmic modification: This procedure is oriented towards the adaptation of base learning methods to be more attuned to class imbalance issues.
- Cost-sensitive learning: This type of solutions incorporate approaches at the data level, at the algorithmic level, or at both levels combined, considering higher costs for the misclassification of examples of the positive class with respect to the negative class, and therefore, trying to minimize higher cost errors.

Into classification task, accuracy is the most used empirical measure, but it is not adequate to the evaluation in imbalanced domains due to the fact that it does not distinguish between the number of correct labels of different classes, which in the framework of imbalanced problems may lead to erroneous conclusions. One of the most used metrics in imbalanced data sets is the geometric mean (GM) of the true rates [3], defined as:

$$GM = \sqrt{\frac{TP}{TP + FN} \frac{TN}{FP + TN}} \tag{1}$$

where TP, TN, FP and FN stand for True Positives, True Negatives, False Positives and False Negatives respectively. This metric attempts to maximize the accuracy of each one of the two classes with a good balance.

3 Cost-Sensitive Learning

Generally speaking, cost-sensitive learning states different costs of misclassification with respect to the existing classes in a problem. With this aim, a cost matrix $C(i, j)$ defines the penalties of classifying examples of one class i as a different one j. Typically, these misclassification costs can be heuristically established or extracted from domain experts. In binary imbalance problems $C(+, -)$ is the cost of misclassifying a positive (class) instance as the negative and $C(-, +)$ is the opposite cost. As mentioned, in this kind of problem, it is more interesting to obtain a correct classification of the positive instance and therefore $C(+, -) > C(-, +)$. According to [19], cost-sensitive learning algorithms can be divided into three categories:

- Direct methods: These algorithms introduce misclassification costs into the learning algorithms. In decision tree algorithms the cost information is used

to choose the best attribute to split the data [18] or to determine whether a subtree should be pruned [4]. In [30] an approach based on genetic algorithms that incorporates misclassification costs in the fitness function is presented. [16] modifies the learning weight algorithm for training RBFNs in order to introduce cost sensitivity. In [28] and [31] a cost-sensitive learning for SVM is implemented, the main idea being to assign a larger penalty value to false negatives than false positives.

– Meta-learning: In this category an additional stage is introduced where the training data are pre-processed or the output is post-processed, remaining the original learning algorithm unmodified. Furthermore, cost-sensitive algorithms can be classified into thresholding and sampling strategies. In thresholding strategies instances are assigned to the class with minimum expected costs as in [9]. In [36] a threshold-moving method is defined with the aim of moving the output threshold toward inexpensive classes such that examples with higher costs become harder to be misclassified. On the other hand, the sampling methodology is based on undersampling/oversampling [35] or assigning instance weights [29]. In [29] an instance-weighting method is defined in order to design cost-sensitive trees.

4 CO²RBFN: An Evolutionary Cooperative-Competitive Hybrid Algorithm for RBFN Design

Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network (ANN) paradigms in the machine design field. An RBFN is a feed-forward ANN with a single layer of hidden units, called radial basis functions (RBFs) [5,21].

From a structural point of view, an RBFN is a feed-forward neural network with three layers: an input layer with n nodes, a hidden layer with m neurons or RBFs, and an output layer (Figure 1).

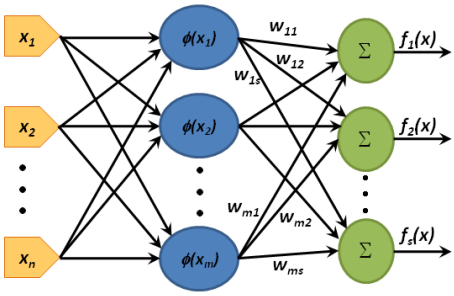


Fig. 1. RBFN Topology

The m neurons of the hidden layer are activated by a radially-symmetric basis function, $\phi_i : \mathbb{R}^n \rightarrow \mathbb{R}$, which can be defined in several ways [27], the Gaussian function being the most widely used (Equation 2):

$$\phi_i(\boldsymbol{x}) = \phi_i(e^{-(\|\boldsymbol{x}-\boldsymbol{c_i}\|/d_i)^2}) \tag{2}$$

where $\boldsymbol{c_i} \in R^n$ is the center of basis function ϕ_i, $d_i \in R$ is the width (radius), and $\|\|$ is typically the Euclidean norm on R^n. This expression is the one used in this paper as the Radial Basis Function (RBF). The output node implements the following function, where weights w_{ij} show the contribution of an RBF to the output node (Equation 3):

$$f_j(\boldsymbol{x}) = \sum_{i=1}^{m} w_{ij}\phi_i(\boldsymbol{x}) \tag{3}$$

The objective of any RBFN design process [6] is to determine centers, widths and the linear output weights connecting the RBFs to the output neuron layer.

An important paradigm for RBFN design is Evolutionary Computation [14]. There are different proposals in this area with different scheme representations: Pittsburgh [15], where each individual is a whole RBFN, and cooperative-competitive [32], where an individual represents a single RBF.

CO²RBFN [25] is an evolutionary cooperative-competitive hybrid algorithm for the design of RBFNs. The network represents the entire population and each individual is represented by a neuron or RBF. The fitness of each individual is known as credit assignment and it is calculated by using three factors: the RBF contribution to the network output, the error in the basis function radius, and the degree of overlapping among RBFs.

The three parameters used for credit assignment are given as input of a Fuzzy Rule-Based System that determines which of the four evolutive operator must be applied to the individual.

The main steps of CO²RBFN, explained in the following subsections, are shown in the pseudocode, in Algorithm 1. For a wider explanation of the algorithm see reference [25].

Algorithm 1. Main steps of CO²RBFN

```
1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the eliminated RBFs
6. Select the best RBFs
7. If the stop condition is not verified go to step 2
```

In the RBFN initialization step, to define the initial network, a specified number m of neurons (i.e. the size of population) is randomly allocated to the

different patterns of the training set. To do so, each RBF center, $\boldsymbol{c}_i$, is randomly assigned to a pattern of the training set. The RBF widths, d_i, will be set to half of the average distance between the centers. Finally, the RBF weights, w_{ij}, are set to zero.

In the RBFN training step, LMS training algorithm is used.

For the RBF evaluation, a credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters:

- The contribution of the RBF, a_i, is determined by considering the weight and the number of patterns of the training set inside its width.
- The error measure, e_i, for each RBF, is obtained by counting the wrongly classified patterns inside its radius.
- The overlapping of the RBF with respect to the other RBFs.

In CO^{2}RBFN four operators have been defined in order to be applied to the RBFs:

- Operator *Remove*: eliminates an RBF.
- Operator *Random Mutation*: randomly modifies the coordinates of the center and the width of an RBF.
- Operator *Biased Mutation*: modifies the width and the coordinates of the center using local information of the RBF environment.
- Operator *Null*: in this case all the parameters of the RBF are maintained.

The operators are applied to the whole population of RBFs. The probability of choosing an operator is determined by means of a Mandani-type fuzzy rule based system [20].

The inputs of this system are parameters a_i, e_i and o_i used for defining the credit assignment of the RBF ϕ_i. These inputs are considered linguistic variables va_i, ve_i and vo_i. The outputs, p_{remove}, p_{rm}, p_{bm} and p_{null}, represent the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively.

The rule base system aims to evolve RBFs with a good behavior (high contribution, low error and low overlapping) and to eliminate RBFs with a bad behavior (low contribution, high error and high overlapping).

In the step of introduction of new RBFs, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the center of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively.

The width of the new RBF will be set to the average of the RBFs in the population plus half of the minimum distance to the nearest RBF. Its weights are set to zero.

The replacement scheme determines which new RBFs (obtained before the mutation) will be included in the new population. To do so, the role of the mutated RBF in the net was compared with the original one to determine the RBF with the best behavior in order to include it in the population.

5 CO²RBFN-CS: CO²RBFN with Cost-Sensitivity

As mentioned in previous sections, approaches that deal with imbalanced data sets can be categorized into three categories: Data sampling, Algorithmic modification and Cost-sensitive. This last approach can incorporate modifications at the data level, at the algorithmic level, or at both levels combined, in order to introduce higher costs for the misclassification classes.

A first base to design CO²RBFN-CS, the adaptation of CO²RBFN for addressing imbalance problems, is that the Positive class is poorly represented (with only a few instances) into an imbalanced data set. CO²RBFN learns the classes of the data set using the Least Mean Square (LMS, [33]) and, in some way, each class is learned from its instances. Nevertheless, this can be a difficult task for the Positive class which can have few instances in an imbalanced data set.

Although the classical LMS algorithm, as it was seen in [24], obtains good performance in imbalanced data sets scenarios, the objective of this paper is to modify it in order to achieve better results. LMS is a local weights training algorithm that uses the gradient descent technique. This technique exploits the local information that can be obtained from the behavior of each RBFs. Equation 4 shows the update of the weights.

$$\overline{w}_{k+1} = \overline{w}_k + \alpha \frac{e_k \phi_i(\overline{x}_k)}{|\phi_i(\overline{x}_k)|^2} \tag{4}$$

where k is the number of iterations, $\overline{w}_{k+1}$ is the next value of the weight vector, $\overline{w}_k$ is the present value of the weight vector and $\overline{x}_k$ is the value of the actual input pattern vector.

The present linear error, e_k, is defined as the difference between the desired output and the output network before adaptation. The α value is the *speed of learning*, which measures the size of the adjustment to be made. The choice of α controls stability and speed of convergence.

The challenge in the new proposal is to redefine the LMS algorithm incorporating cost-sensitivity. Few methods have been proposed in this line, the most similar is [16] where cost-sensitivity is introduced in the weights training algorithm, and although it is a gradient descent based algorithm, it does not use the classic LMS algorithm. Moreover [16] is designed to solve an specific problem.

Focusing on our proposal, as can be seen in equations 5, a new variable *cost* is introduced in the LMS algorithm. The equations 6, 7 are used to explain this cost, instead of a cost matrix, because it depends on three aspects: the IR of the data set, the classified class (Positive or Negative) and the current success on the classification of the instance. Obviously *cost* depends on the classified class and in this way its value is mainly modified when a pattern of the minority class is learned. The IR of the dataset is taken into account because when this ratio is high, that means there are few positive patterns, it is necessary to increase the learning of rate of these patterns in order to improve the accuracy. Finally, if an instance is being wrongly classified its learning rate will be increased.

In summary, the *cost* is higher if the IR is higher than 9, the classified class is the minority one and the instance is being wrongly classified by the model.

On the other hand the *cost* is lower if the IR is lower than 9, the classified class is the majority one and the instance is being correctly classified by the model.

$$\overline{w}_{k+1} = \overline{w}_k + cost * \alpha \frac{e_k \phi_i(\overline{x}_k)}{|\phi_i(\overline{x}_k)|^2} \tag{5}$$

$$cost = \begin{cases} csuccess & \text{well classified minority class pattern} \\ cerror & \text{wrong classified minority class pattern} \\ 1 & \text{majority class pattern} \end{cases} \tag{6}$$

$$\begin{aligned} IR < 9 \begin{cases} csuccess = 1 \\ cerror = 2 \end{cases} \\ IR >= 9 \begin{cases} csuccess = 2 \\ cerror = 4 \end{cases} \end{aligned} \tag{7}$$

6 Experimentation and Results

The methodology to test CO^{2}RBFN-CS has consisted in comparing it with its base version CO^{2}RBFN and after that, with other more mature cost-sensitive methods. With this objective, different data sets have been chosen from the KEEL data set repository [1]. Table 1 summarizes the data employed in this study and shows, for each data set, the number of examples (#Ex.), number of attributes (#Atts.), class name of each class (minority and majority), class attribute distribution and IR. This table is ordered by the IR, from low to highly imbalanced data sets. Half of the data sets have an IR lower than 9 and the remaining ones have an IR higher than 9.

Table 1. Description for imbalanced data sets

Data sets	#Ex.	#Atts.	Class(min., maj.)	%Class(min., maj.)	IR
glass0	214	9	(build-win-float-proc, remainder)	(32.71, 67.29)	2.06
haberman	306	3	(Die, Survive)	(27.42, 73.58)	2.68
glass0123vs456	214	9	(non-window glass, remainder)	(23.83, 76.17)	3.19
vehicle0	846	18	(Van, remainder)	(23.64, 76.36)	3.23
ecoli2	336	7	(pp, remainder)	(15.48, 84.52)	5.46
pageblocks0	5472	10	(remainder, text)	(10.23, 89.77)	8.77
wowel0	988	13	(hid, remainder)	(9.01, 90.99)	10.10
shuttle0vs4	1829	9	(Rad Flow, Bypass)	(6.72, 93.28)	13.87
yeast2vs8	482	8	(pox, cyt)	(4.15, 95.85)	23.10
yeast1289vs7	947	8	(vac, nuc, cyt, pox, erl)	(3.17, 96.83)	30.56
yeast5	1484	8	(me1, remainder)	(2.96, 97.04)	32.78
yeast6	1484	8	(exc, remainder)	(2.49, 97.51)	39.15

With these data sets, a typical experimental framework has been established with 5-fold cross validation and five repetitions for obtaining the results.

Firstly the new proposal, CO^{2}RBFN-CS, and the base version, CO^{2}RBFN, are compared. The same configuration parameters are set up for the two algorithms: 200 iterations are established for the main loop and the number of individuals or RBFs is set to five.

In Table 2 the average correct classification rates, using GM measures, obtained by both versions (the original CO^{2}RBFN and the CO^{2}RBFN-CS) are shown.

Table 2. Average GM test results

Data sets	CO^{2}RBFN-CS	CO^{2}RBFN
glass0	78.314	75.693
haberman	63.206	61.209
glass0123vs456	93.363	92.269
vehicle0	90.776	89.116
ecoli2	92.072	92.024
pageblocks0	87.832	86.073
vowel0	91.717	87.026
shuttlec0vsc4	99.665	99.671
yeast2vs8	76.291	71.883
yeast1289vs7	68.788	55.190
yeast5	94.803	94.121
yeast6	87.840	83.269

Table 3. Wilcoxon test. R^+ corresponds to new proposal, CO^{2}RBFN-CS, and R^- to CO^{2}RBFN

R^+	R^-	p-value
77.0	1.0	0.002526

If these first results are analyzed (Table 2), it can be observed that CO^{2}RBFN-CS outperforms CO^{2}RBFN in 11 of the 12 tested data sets.

In order to detect significant differences, the Wilcoxon signed-ranks test [34] is applied to compare the results of each version. Table 3 shows the result of the test, in this table R+ corresponds to CO^{2}RBFN-CS and and R- to CO^{2}RBFN. The p-value obtained is very low, indicating that the null hypothesis of equality of means is rejected with a high confidence level. Therefore, CO^{2}RBFN-CS outperforms CO^{2}RBFN with significant difference.

Focusing the analysis on the IR of the data sets, we can observe that when the IR is higher than 9 the difference between the methods grows. As example, for yeast6 CO^{2}RBFN-CS beats CO^{2}RBFN by 4 points and for yeast1289vs7 by 13 points. These results indicate that the cost-sensitive methodology proposed in this paper is even more suitable when the IR grows.

Now, CO^{2}RBFN-CS is compared with other more mature cost-sensitive methods, specifically with the cost-sensitive methods implemented in Keel, these are:

Table 4. Average GM test results

Data sets	CO^2RBFN-CS	SVM-CS-RBF	C4.5-CS	NN-CS
glass0	78.314	78.799	78.450	59.969
haberman	63.206	52.040	50.040	61.541
glass0123vs456	93.363	93.624	87.459	91.191
vehicle0	90.776	40.095	92.850	65.587
ecoli2	92.072	90.764	72.263	67.314
pageblocks0	87.832	60.895	97.957	70.522
vowel0	91.717	100.000	88.702	66.653
shuttlec0vsc4	99.665	95.003	99.971	84.736
yeast2vs8	76.291	75.269	84.698	65.190
yeast1289vs7	68.788	69.324	62.057	43.873
yeast5	94.803	96.931	93.098	62.166
yeast6	87.840	87.406	85.337	55.909

Table 5. Friedman ranking test

Algorithm	Ranking
CO^2RBFN-CS	1.833
SVM-CS-RBF	2.167
C4.5-CS	2.417
NN-CS	3.583

Table 6. Hochberg post hoc test

i	algorithm	$z = (R_0 - R_i)/SE$	p	Hochberg
3	NN-CS	3.320392	0.000899	0.016667
2	C4.5-CS	1.106797	0.268382	0.025
1	SVM-CS-RBF	0.632456	0.527089	0.05

- SVM-CS-RBF [28,31]: a cost-sensitive learning for SVM, with RBF Kernel, that assigns a larger penalty value to false negatives than false positives.
- C4.5-CS [29]: an instance-weighting method to design cost-sensitive trees.
- NN-CS [36]: a threshold-moving method that aims to move the output threshold toward inexpensive classes such that examples with higher costs become harder to be misclassified.

Further information can be found in section 3 and in the referenced bibliography. The parameters established for these methods are the default ones configured in Keel. The results obtained for the different methods are shown in Table 4.

With the objective of carrying out an adequate multiple comparison, as recommended in [13], the statistical analysis was performed in two steps. Firstly, the Friedman test [12] is used to rank the methods, and to establish if any

statistical differences exist. In the ranking of the methods obtained by this test, Table 5, CO^{2}RBFN-CS achieves the best (lowest) value. The p-value computed by Friedman test is 0.0058, which indicates significant differences among the methods. Secondly, in order to elucidate these significant differences between CO^{2}RBFN-CS and the remaining methods, the Hochberg test [17] has been applied, obtaining the results showed in Table 6. Taking CO^{2}RBFN-CS as control algorithm, because is the best algorithm according to Friedman test, it can be highlighted that CO^{2}RBFN-CS obtains significant differences with respect to a more mature method such as NN-CS, and a notable p-value with respect to C4.5-CS. In any case it is remarkable that this first approach introducing cost-sensitivity into a cooperative-competitive design of RBFNs, CO^{2}RBFN-CS, has achieved similar or even better results than other more tested methods.

7 Conclusions

Currently, the data sets that represent an important number of real applications are imbalanced. These imbalance problems are addressed with different methods which can be categorized into three groups. data sampling, algorithmic modification and cost-sensitive learning. Cost-sensitive learning can hybridize both data sampling and algorithmic modification in order to incorporate higher costs for the misclassification of examples, trying to minimize higher cost errors.

In this paper CO^{2}RBFN-CS, a first approach in the cooperative-competitive design of RBFNs with cost-sensitivity, is presented. This proposal is based on introducing a cost parameter into the training weights stage of the algorithm in order to modifying its learning rate. This costs depends on the right (or wrong) classification of the instance, the type of class (majority or minority) and the imbalance ratio of the data set.

The results show that CO^{2}RBFN-CS outperforms CO^{2}RBFN in 11 of the 12 tested data sets and after applying Wilcoxon's test, significant differences have been achieved. Also, it can be observed that CO^{2}RBFN-CS even works better when the IR of the data set is higher.

When CO^{2}RBFN-CS is compared with more mature cost-sensitive methods in the bibliography, it can be highlighted that CO^{2}RBFN-CS achieves the first position in the ranking of the Friedman and, even, outperforms the NN-CS method with significant differences. In summary, this first approach in the cost-sensitive cooperative-competitive design of RBFNs, CO^{2}RBFN-CS, has obtained promising results, similar or even higher than other more mature methods. As future research lines the proposed equation will be modified adjusting the value of the parameters or introducing new ones related with the neighborhood of positive patterns.

Acknowledgments. F. Charte is supported by the Spanish Ministry of Education under the F.P.U. National Program (Ref. AP2010-0068). This paper is partially supported by the Spanish Ministry of Science and Technology under the Project TIN 2012-33856, FEDER founds.

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. of Mult.-Valued Logic & Soft Computing 17, 255–287 (2011)
2. Alexandridis, A., Chondrodima, E., Sarimveis, H.: Radial basis function network training using a nonsymmetric partition of the input space and particle swarm optimization. IEEE Transactions on Neural Networks and Learning Systems **24**(2), 219–230 (2013)
3. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition **36**(3), 849–851 (2003)
4. Bradford, J.P., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.E.: Pruning decision trees with misclassification costs. In: Proceedings of the the 10th European Conference on Machine Learning (ECML 1998), pp. 131–136 (1998)
5. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex Systems **2**, 321–355 (1988)
6. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE Transactions on System, Man, and Cybernetics, B **35**(5), 928–947 (2005)
7. Chawla, N.V., Japkowicz, N., Kolcz, A.: Special issue on learning from imbalanced data sets. SIGKDD Explorations Newsletter **6**(1), 1–6 (2004)
8. Chen, H., Kong, L., Leng, W.: Numerical solution of pdes via integrated radial basis function networks with adaptive training algorithm. Applied Soft Computing **11**, 856–860 (2011)
9. Domingos, P.: Metacost: a general method for making classifiers cost sensitive. In: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, pp. 155–164 (1999)
10. Du, H., Zhang, N.: Time series prediction using evolving radial basis function networks with new enconding scheme. Neurocomputing **71**, 1388–1400 (2008)
11. Fernández, A., del Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification system with genetic rule selection for imbalanced data-set. International Journal of Approximate Reasoning **50**, 561–577 (2009)
12. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association **32**, 675–701 (1937)
13. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences **180**, 2044–2064 (2010)
14. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
15. Harpham, C., Dawson, C.W., Brown, M.R.: A review of genetic algorithms applied to training radial basis function networks. Neural Computing and Applications **13**, 193–201 (2004)
16. He, Z.M.: Cost-sensitive steganalysis with stochastic sensitvity and cost sensitive training error. Proceedings of the International Conference on Machine Learning and Cybernetics **1**, 349–354 (2012)
17. Hochberg, Y.: A sharper bonferroni procedure for multiple tests of significance. Biometrika **75**(4), 800–802 (1988)

18. C. X. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In: Proceedings of the 21st International Conference on Machine Learning (ICML 2004), vol. 69, pp. 544–551 (2004)
19. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classiffcation with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Information Sciences **250**, 113–141 (2013)
20. Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies **7**(1), 1–13 (1975)
21. Moody, J., Darken, C.J.: Fast learning in networks of locally-tuned processing units. Neural Computation **1**, 281–294 (1989)
22. Niu, H.L., Wang, J.: Financial time series prediction by a random data-time effective RBF neural network. Soft Computing **18**(3), 497–508 (2014)
23. Pérez-Godoy, M.D., Fernández, A., Rivera, A.J., del Jesus, M.J.: Analysis of an evolutionary RBFN design algorithm, CO^{2}RBFN, for imbalanced data sets. Pattern Recognition Letters **31**(15), 2375–2388 (2010)
24. Pérez-Godoy, M.D., Rivera, A.J., Carmona, C.J., del Jesus, M.J.: Training algorithms for radial basis function networks to tackle learning processes with imbalanced data-sets. Applied Soft Computing **25**, 26–39 (2014)
25. Pérez-Godoy, M.D., Rivera, A.J., del Jesus, M.J., Berlanga, F.J.: CO^{2}RBFN: An evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft Computing **14**(9), 953–971 (2010)
26. Qasem, S.N., Shamsuddin, S.M.: Memetic elitist pareto differential evolution algorithm based radial basis function networks for classification problems. Applied Soft Computing **11**(8), 5565–5581 (2011)
27. Rojas, I., Valenzuela, O., Prieto, A.: Statistical analysis of the main parameters in the definition of radial basis function networks. LNCS **1240**, 882–891 (1997)
28. Tang, Y., Zhang, Y.-Q., Chawla, N.V., Krasser, S.: Svms modeling for highly imbalanced classification. IEEE Transactions on Systems Man and Cybernetics. PART B. Cybernetics **39**(1), 281–288 (2009)
29. Ting, K.M.: An instance-weighting method to induce cost-sensitive trees. IEEE Transactions on Knowledge and Data Engineering **14**(3), 659–665 (2002)
30. Turney, P.D.: Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of Artificial Intelligence Research **2**, 369–409 (1995)
31. Veropoulos, K., Cristianini, N., Campbell, C.: Controlling the sensitivity of support vector machines. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 1999), pp. 55–60 (1999)
32. Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. IEEE Transactions on Neural Networks **7**(4), 869–880 (1996)
33. Widrow, B., Lehr, M.A.: 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proceedings of the IEEE **78**(9), 1415–1442 (1990)
34. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics **1**, 80–83 (1945)
35. Zadrozny, B., Langford, J., Abe, N.: Costsensitive learning by costproportionate-example weighting. In: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), pp. 435–442 (2003)
36. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge Data Engineering **18**(1), 63–77 (2006)

Transfer Learning for the Recognition of Immunogold Particles in TEM Imaging

Ricardo Gamelas Sousa[5,8], Tiago Esteves[1,5,8], Sara Rocha[2], Francisco Figueiredo[4,8], Joaquim Marques de Sá[5,8], Luís A. Alexandre[6(✉)], Jorge M. Santos[5,7,8], and Luís M. Silva[3,5,8]

[1] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
[2] Centro de Biotecnologia dos Açores (CBA), Universidade dos Açores, Açores, Portugal
sararocha@uac.pt
[3] Departamento de Matemática at Universidade de Aveiro, Aveiro, Portugal
[4] Instituto de Biologia Molecular e Celular (IBMC), Porto, Portugal
[5] Instituto de Engenharia Biomédica (INEB), Porto, Portugal
[6] Instituto Telecomunicações (IT), Universidade da Beira Interior, Covilhã, Portugal
lfbaa@di.ubi.pt
[7] Instituto Superior de Engenharia, Politécnico do Porto, Porto, Portugal
[8] Instituto de Investigação e Inovação em Saúde, Universidade do Porto, Porto, Portugal
rsousa@rsousa.org, lmas@ua.pt, jms@isep.ipp.pt, dee11017@fe.up.pt, francisco.figueiredo@ibmc.up.pt

Abstract. We present a Transfer Learning (TL) framework based on Stacked denoising Autoencoder (SDA) for the recognition of immunogold particles. These particles are part of a high-resolution method for the selective localization of biological molecules at the subcellular level only visible through Transmission Electron Miscroscopy (TEM). Four new datasets were acquired encompassing several thousands of immunogold particles. Due to the particles size (for a particular dataset a particle has a radius of 4 pixels in an image of size 4008×2670) the annotation of these datasets is extremely time taking. Thereby, we apply a (TL) approach by reusing the learning model that can be used on other datasets containing particles of different (or similar) sizes. In our experimental study we verified that our (TL) framework outperformed the baseline (not involving TL) approach by more than 20% of accuracy on the recognition of immunogold particles.

1 Introduction

Common Machine Learning create new classifiers whenever the respective probability distributions of inputs and outputs change, even though they may relate to similar problems. The *reuse* of a classifier designed for a given (*source*) problem on another (*target*) problem, presenting some similarities with the original one, with only minor operations of parameter tuning, is the scope of Transfer Learning (TL).

This paper proposes a TL framework for the recognition of immunogold particles on datasets with different magnifications. The structure of this framework is

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 374–384, 2015.
DOI: 10.1007/978-3-319-19258-1_32

as follows: (i) a Stacked Denoising Autoencoder (SDA) model is built to recognize particles of a specific size; (ii) in order to ease the burden of image annotation and aiming for an effective particle recognition a robust SDA obtained through TL is developed. To the best of our knowledge this work is the first TL framework of its kind for immunogold particle recognition. Furthermore, we also make publicly available a new dataset that can be used as benchmark on future works.

Immunogold electron microscopy is a high-resolution method for the selective localization of biological molecules at the subcellular level. Antibodies coupled to particles of colloidal gold, which are visible through Transmission Electron Microscopy (TEM), can reveal the localization and distribution of the biological molecules of interest. In this particular work, this technique was used to determine the composition of cell wall uneven thickenings that ultimately differentiate into reticulate and flange ingrowths of maize (*Zea mays* L.) endosperm transfer cells [15]. These cells are essential for assimilate flow into the endosperm, thus having a significant impact on kernel yield.

Immunogold particle counting is a time-consuming task where a single image containing almost a thousand particles can take several hours to annotate [19] (Figure 1). Moreover, recent technological improvements in microscopy have made easier and faster to acquire large volumes of data of biological samples at very high, Electron Microscopy (EM)-level, magnifications. To add, images are usually recorded with different magnifications for a better analysis of the biological tissues. Conducting this analysis on a daily basis is not only a tiresome task but is also very time consuming. An automatic recognition tool can reduce the time consumed in such analysis and improve its accuracy (by removing the bias of a manual annotation) thus being of paramount importance. Ultimately, through TL we can relax the necessity of obtaining large amounts of labeled data for new problems [2,5] (such as new magnifications); and, less computational effort by providing new learning models that can be applied with good performance results in different problems and far less computational effort (see e.g., [5,10,11]).

This paper is structured in the following way: we start by briefly describing previous works on the recognition of cellular structures aided by automatic systems (Section 2). In Section 3 we present our TL framework for the recognition of immunogold particles. In Section 4 we present the results and the conclusions are drawn in Section 5.

2 Previous Work

Fisker *et al.* in [8] explores the possibility to automatically estimate particle sizes in immuno-microscopy imaging. Their approach is based on deformable models that can be fitted to the prior known shape of the particles. With the same goal as Fisker, a different approach was presented by Mallic *et al.* in [13] by using cascade of classifiers. Inspired in the conventional face detection challenges where recognition with ensemble of weak classifiers have proven to be effective, this straightforward strategy may be computationally complex due to the amount of immunogold particles that may occur in electron microscopy images. In [19]

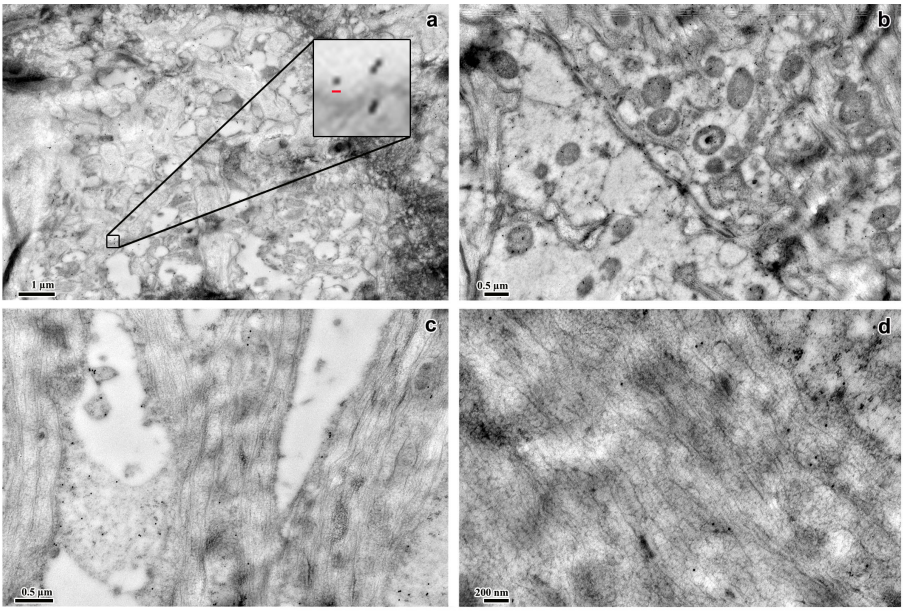


Fig. 1. Representative images of our datasets illustrating different structures that can interfere in the recognition of the immunogold particles due to: cellular overlapping, tissues and background noise. Each image has 4008×2670 pixels of dimension with particles *diameter* ranging from 8 to 20 pixels. **(a)** Example of a sample with a magnification of 15000 ($1\mu m$, particles with a diameter of 8 pixels—red line); **(b)** magnification of 20000 ($0.5\mu m$, 12 pixels diameter particles); **(c)** magnification of 30000 ($0.5\mu m$, 15 pixels diameter particles); and, **(d)** magnification of 50000 ($200nm$, 20 pixels diameter particles).

an insight review is presented by motivating the advantages of Computer Vision (CV) for the processing of images generated by electron microscopies. Finally, in [23,24] a Difference of Gaussian (DoG) and Laplacian of Gaussian (LoG) filters are applied to aid the detection of particles (e.g., organelles) on cryo-electron microscopy images. On these papers, DoG or LoG were used as a first step to detect more complex biological structures and were not tailored neither evaluated on immunogold particles. For the *detection* of biological structures, there is the publicly available Spot Detector (SD) [17] algorithm that is included in the well-known Icy bioimaging software [7]. Authors state that SD enables the detection of spots that can be organelles or other biological similar structures in noisy 2D or 3D images [17].

A major difference is that in the aforementioned proposals all particle structures were shallow, irregular in shape and intensity. Our work will be focused on the recognition of immunogold particles with regular spherical shape, thus avoiding the adoption of a highly parameterized formalism for its recognition. Furthermore, the creation of different models, each one developed for a specific

problem, can be cumbersome due to the time that is needed to label the data. For this purpose TL is presented as an appealing alternative.

TL has been around since the 80's with considerable advancements since then (see e.g., [9,14,18,21] and references therein). With the recent re-interest on Neural Networks (NNs) and the availability of more computational power along with new and faster algorithms, NNs with deep architectures started to emerge to tackle TL. In [12] (and work referenced therein) the authors have widened the research line on TL by addressing the following questions: How can one tailor Deep Neural Networks (DNNs) for TL? How does TL perform by reusing layers and using different types of data? For the application of Microscopy Imaging a TL categorization was presented in [4] for the recognition of mitochondria. With the exception of [16] where an adaboost and TL framework is proposed for the detection of cells with different features (e.g., shape, size), there are very few works that address the possibility of TL on biomedical data, specifically for the recognition of immunogold particles.

Current approaches are not tailored for immunogold particle detection and recognition and thus are prone to error by erroneously estimating a microstructure organelle or an artifact as immunogold particle due to the criteria of the filter responses: circular shape and radius dimension. To this end a robust mechanism is required to handle possible false detections that can occur in this kind of images. In the following Sections we will present a SDA-based TL mechanism as an effective approach for immunogold particle recognition.

3 A TL Framework for the Recognition of Immunogold Particles Using SDAs

Our proposal comprises the following two steps:

- a SDA is first trained to distinguish immunogold particles from cluttered background;
- the (learning) model is then reused onto another dataset (with particles of a different size).

In the following sections we will thoroughly describe the methods (for a self-contained reading) and the framework here proposed.

3.1 SDAs for the Recognition of Immunogold Particles

An autoencoder is a simple NN with one hidden layer designed to reconstruct its own input, having, for that reason, an equal number of input and output neurons. The reconstruction accuracy is obtained by minimizing the average reconstruction error between the original and the reconstructed instances. Hence, these methods are governed by the objective of capturing relevant information of the underlying distribution of the samples [20]. The encoding and decoding feature sets (input-hidden and hidden-output weights, respectively) may optionally be constrained as transpose of each other, in which case the autoencoder is

said to have tied weights. For a variant of the autoencoders, Denoising Autoencoder (DAE), input data is corrupted to train a model in order to rebuild the original data. Such is achieved by means of the minimization of some reconstruction loss and allows the DAEs to avoid a direct copy of the data [22]. Stacking autoencoders, gives the model the advantage of hierarchical features with low-level features represented at lower layers and higher-level features represented at upper layers [6,22, Section3]. The unsupervised training of the SDAs is usually referred to as pre-training. We then add to the top of the network a logistic layer and the entire network is "fine-tuned" in order to minimize some classification loss function [1,6].

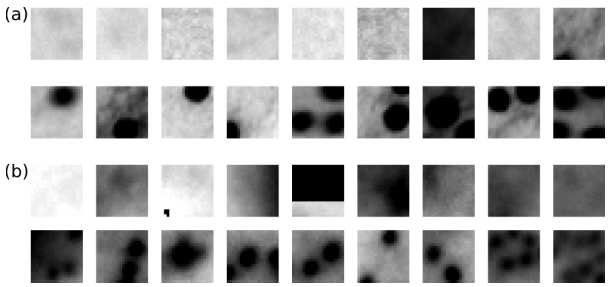


Fig. 2. Different types of patches in the dataset with magnification of 50000 (a): background and artifact patches (two top rows); and, patches containing at least one immunogold particle (two bottom rows); and, (b) analogous for the patches in the dataset with magnification of 15000

SDAs robustness makes it a very promising learning tool for the recognition of the circular shaped immunogold particles. A representative sample of the images that were used to train the SDA is depicted in Figure 2. Given the similarity of the immunogold particles, it is expected that a SDA can capture relevant features from these samples and easily discriminate from the remaining artifacts or cellular structures. A far more complex scenario occurs when multiple immunogold particles are comprised in the same patch (see Figure 2), especially in low magnifications such as the magnification of 15000. In this case the SDA has to be able to deal with a variable number of particles in the same image. Besides this, it also has to be robust to the different particle sizes.

Although it may seen unsual to capture images in different magnifications, the automatic analysis in lower magnifications (magnification of 15000) helps a direct identification of regions of interest and its direct quantification as opposed to when using higher magnifications. By directly exploring images obtained at lower magnifications we can attain reduced experimentation costs, image acquisition and labelling times. However, the performance of an automatic learning model for these images is prone to misclassify a great number of images due to the noise and artifacts present in the dataset. For this, we can build a learning model using a dataset from higher magnifications, such as magnification of 50000

(containing immunogold particles well defined), and transfer it to a magnification of 15000.

Therefore, a TL setting for the recognition of immunogold particles without requiring the learning of new models from scratch (e.g., by initializing the layer weights at random) is important.

3.2 TL: Reusability of SDA Models

TL aims to transfer models acquired in one problem, the source problem, onto another problem, the target problem, dispensing the bottom-up construction of the target model. TL algorithms embedded in DNNs have already shown interesting features such as better generalization capabilities and reduced training times as shown in other works [10] (and references therein). In here we extend these studies of model reusability to the recognition of immunogold particles in images of maize endosperm transfer cells. A dataset for a given magnification is identified as the source problem and the (source) learning model will be built upon this data. TL is then applied by reusing the source model on the target problem (in this case, a dataset with a different magnification). Putting it simply, layer weights for the target model are initialized with the values of the source model. Learning (of the target problem) can be conducted in several ways: (a) by fixing the layer weights of the network or let them readjust through the minimization of the reconstruction error (also known as pre-training); (b) by fixing or letting the network relearn and thus letting it readjust the decision function (fine-tune); or (c) a mixture of both [2,3,6,10,11] (and references therein).

For the purpose of this work, we have reused the source model on the target problem by letting it readjust the decision function to the target problem. In our experimental work we have analyzed the impact of the reusability of the different hidden layers. In fact, and for the immunogold particle recognition, one may argue that the major changes will occur at a feature level thanks to the changes of the magnification of the immunogold particles. Looking carefully at Figure 1 we can see that this seems to be the case. Therefore, the recognition performance of the immunogold particles through TL would have significant impacts by relearning only the first layers of the network. The analysis of the effects of relearning specific parts of the network architecture will be addressed in more detail in Section 4.

4 Methodology and Results Discussion

4.1 Dataset

All images were acquired using a Transmission Electron Microscope JEOL JEM 1400 with a GATAN Orius SC10000A2 CCD and were recorded in four different magnifications: 15, 20, 30 and 50 thousand times from different biological samples. See Figure 1 for a brief illustration of this dataset. For each magnification we recorded 25 digital images. Each image contains 500 to 200 particles annotated by a trained expert (localization on the image of an immunogold particle).

4.2 Experimental Study

Dataset for SDA Models Construction: For a given image we extracted *all* immunogold particles and the same amount of background, cellular structures or artifacts patches with 20×20 pixels (the latter at random positions). A patch could contain more than one particle or portions of several other particles (see Figure 2). Finally, patches were labeled as containing at least one immunogold particle if the Euclidean distance between the patch central position (on the image) and the annotation position was below 10 pixels (half of the size of the patch width). Moreover, we ensure that there is a one-to-one mapping between the patch location and the ground-truth.[1] This procedure thus results on a dataset (for a given magnification) containing several thousand patches of immunogold particles and background, cellular structures or artifacts.

Training Procedure: The experimental procedure for the construction of the baseline models was conducted as follows. For a given resolution we randomly split the data into training and test data: 60% to train our SDA (60 images in total, 15 samples per resolution) and the remaining for testing purposes. Pixel intensity values of all patches were normalized to be within $[0, 1]$. To find the best SDA model parameterization we have performed a grid search on the pre-training learning rate (0.01 and 0.001) and fine-tuning learning rate (0.1 and 0.01) by carrying out a three-fold cross-validation in the training set. The input layer had 400 neurons (for handling the patches size of 20×20). The number of hidden layers was fixed to 3 and the number of neurons per layer was fixed to 500. The corruption level was set to 0.1 across all hidden layers. Regarding TL, we proceeded similarly in order to find the best fine-tune learning rate. Due to the different amount of samples, we also performed a cross-validation over the batch-size. Finally, to assess the variability of our methods' performance the experiment was repeated 20 times by randomly shuffling the data.

TL Evaluation: The procedure described above, conducted for a single magnification, allows us to obtain our baseline (without TL) model. For this particular study, such was conducted for the dataset with magnification of 15000 (target problem). Then, we have proceeded with the TL approach. A learning model is trained on the dataset with magnification of 50000 (source problem). Images of this dataset were resized to half (keeping the aspect ratio) of our source problem, but maintaining the size of the patches (20×20). This led to immunogold particles with a radius around 7 pixels (recall that the radius size of immunogold particles on the dataset with magnification of 15000 was 4 pixels). The model obtained in the source problem was then reused for learning the dataset with a magnification of 15000 (target problem).

In our framework, TL can be conducted by reusing the first layer of the model and by fine-tuning only the remaining layers to the target problem (coded

[1] A given manual position will not result on more than one patch.

[011]); another possibility is reusing the first and second layers and fine-tuning the last layer ([001]). By reusing the full network ([000]) the model would not suffer changes, and thus would have not been fine-tuned to the target problem. Results are presented in Table 1.

Table 1. Results of the application of TL to the recognition of immunogold particles. The baseline model was trained in a standard ML way on the dataset with magnification of 15000 (target problem). A model trained for the dataset with magnification of 50000 (source problem) was obtained and it was reused on the target problem. Overall, all TL approaches achieved an improvement of more than 20%. Each layer-wise TL strategy is illustrated in the column Reusability and Fine-Tuning Setting (see main text).

Method	Source	Target	Reusability and Fine-Tuning Setting	Accuracy (± std. dev) (TL)
Baseline	-	15000	-	0.65 ± 0.17
TL	50000	15000	[011]	0.88 ± 0.01
TL	50000	15000	[001]	0.86 ± 0.02
TL	50000	15000	[110]	0.90 ± 0.01
TL	50000	15000	[100]	0.88 ± 0.01
TL	50000	15000	[111]	$\mathbf{0.91 \pm 0.01}$

Discussion: Considering the results shown in Table 1 we see that by applying any of the TL settings we can indeed improve the baseline results (65%). First, we speculate that the reason for the low accuracy rate of the baseline is concerned with the high variability location of the particles. Having smaller sizes (radius of 4 pixels), and given the unconstrained approach of the dataset construction (see Section 3.1), the network will be biased to underperform on the immunogold recognition. Second, the dataset with magnification of 15000 contains a significant number of clusters of immunogold particles increasing the difficulty of processing patches from this dataset. Moreover, this dataset is also comprised of a significant amount of artifacts (see Figure 1). Therefore, by learning a more well defined (source) problem (dataset with magnification of 50000), though still acquired through an unconstrained approach, a better starting point for learning the target problem is provided. Transferring the model and letting all layer weights be fine-tuned ([111]) to the target problem achieved the best result (91%). Another result from our experimental study is concerned with the relevance of the feature representation that the network has captured. From Table 1 we can infer that the first layers are the most relevant. In fact, letting the first two layers of the network be re-learnt and fixing only the last layer (reusability setting [110]) achieved a higher performance (90%) in comparison with only re-learning the last layer ([001] with an accuracy of 86%).

5 Conclusions

Current approaches for the quantification and analysis of histological datasets of transfer cells are essentially manual processes. This procedure is highly time-consuming, prone to error and is suitable for an automated processing by a computer. For the immunogold particles quantification there are no algorithms that claim to be able to automate tasks such as the immunogold labelling. Moreover, recent technological improvements in microscopy have made easier and faster the acquisition of large volumes of data of biological samples at very high, Electron Microscopy (EM)-level, magnifications. Its automatic recognition is therefore of paramount importance.

In this work we have presented a Transfer Learning (TL) framework based on Stacked Denoising Autoencoder (SDA) for automatic classification of immunogold particles with different magnifications. Our results have shown the robustness of the SDA given the amount of noise present in TEM images from Maize samples (see Figure 2). We have also illustrated the robustness of the possible TL settings for the reusability of the networks by attaining accuracy improvements of more than 20% in comparison to the baseline approach. We were also able to show the relevance of the features that were learnt in the first layer so that the network could be tailored to the new (target) problem.

For future work we plan to assess the reusability of the network on other datasets with immunogold particles with more drastic differences of sizes. This would allow us to assess the robustness of the SDA with respect to the invariance of particle size. This will be accomplished by exploring the remaining magnifications. In the future, we will also consider other Deep Neural Network (DNN) architectures.

Acknowledgments. We would like to acknowledge to Fundação para a Ciência e a Tecnologia (FCT) for funding this research through project SFRH/BD/80508/2011, PTDC/EIA-EIA/119004/2010 and PEst-C/SAU/LA0002/2013. Sara Rocha was supported by Grant BIIC M3.1.6/F/038/2009 from Direcção Regional de Ciência e Tecnologia and by Grant SFRH/BD/8122/2002 from FCT. We thank Rui Fernandes from HEMS department for aiding us with the technical knowledge for conducting this work.

References

1. Amaral, T., Silva, L.M., Alexandre, L.A., Kandaswamy, C., Santos, J.M., de Sá, J.M.: Using different cost functions to train stacked auto-encoders. In: 2013 12th Mexican International Conference on Artificial Intelligence (MICAI), pp. 114–120. IEEE (2013)
2. Amaral, T., Silva, L.M., Alexandre, L.M., Kandaswamy, C., de Sá, J.M., Santos, J.: Improving Performance on Problems with Few Labelled Data by Reusing Stacked Auto-Encoders. In: ICMLA (2014)
3. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I.J., Bergeron, A., Bouchard, N., Bengio, Y.: Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop (2012)

4. Becker, C., Christoudias, C., Fua, P.: Domain adaptation for microscopy imaging. IEEE Transactions on Medical Imaging, PP(99), 1–1 (2014)
5. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine Learning **79**(1–2), 151–175 (2009)
6. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. Journal of Machine Learning Research-Proceedings Track **27**, 17–36 (2012)
7. de Chaumont, F., Dallongeville, S., Chenouard, N., Hervé, N., Pop, S., Provoost, T., Meas-Yedid, V., Pankajakshan, P., Lecomte, T., Le Montagner, Y., et al.: Icy: an open bioimage informatics platform for extended reproducible research. Nature methods **9**(7), 690–696 (2012)
8. Fisker, R., Carstensen, J.M., Hansen, M.F.: Bødker, F., Mørup, S.: Estimation of nanoparticle size distributions by image analysis. Journal of Nanoparticle Research **2**(3), 267–277 (2000)
9. Huang, J., Gretton, A., Borgwardt, K.M., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: Advances in Neural Information Processing Systems, pp. 601–608 (2006)
10. Kandaswamy, C., Silva, L.M., Alexandre, L.M., Sousa, R., Santos, J., de Sá, J.M.: Improving transfer learning accuracy by reusing stacked denoising autoencoders. In: Proceedings of the IEEE SMC Conference (2014)
11. Kandaswamy, Chetak, Silva, Luís M., Alexandre, Luís A., Santos, Jorge M., de Sá, Joaquim Marques: Improving Deep Neural Network Performance by Reusing Features Trained with Transductive Transference. In: Wermter, Stefan, Weber, Cornelius, Duch, Włodzisław, Honkela, Timo, Koprinkova-Hristova, Petia, Magg, Sven, Palm, Günther, Villa, Alessandro E.P. (eds.) ICANN 2014. LNCS, vol. 8681, pp. 265–272. Springer, Heidelberg (2014)
12. Kandaswamy, C., Silva, L.M., Alexandre, L.A., Sousa, R., Santos, J.M., de Sá, J.M.: Improving transfer learning accuracy by reusing stacked denoising autoencoders. In: 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 1380–1387. IEEE (2014)
13. Mallick, S.P., Zhu, Y., Kriegman, D.: Detecting particles in cryo-em micrographs using learned features. Journal of Structural Biology **145**(1), 52–62 (2004)
14. Mitchell, T.M.: The need for biases in learning generalizations. Laboratory for Computer Science Research, Rutgers Univ, Department of Computer Science (1980)
15. Monjardino, P., Rocha, S., Tavares, A.C., Fernandes, R., Sampaio, P., Salema, R., da, Câmara Machado, A.: Development of flange and reticulate wall ingrowths in maize (Zea mays L.) endosperm transfer cells. Protoplasma **250**(2), 495–503 (2013)
16. Nguyen, Nhat H., Norris, Eric, Clemens, Mark G., Shin, Min C.: Rapidly Adaptive Cell Detection Using Transfer Learning with a Global Parameter. In: Suzuki, Kenji, Wang, Fei, Shen, Dinggang, Yan, Pingkun (eds.) MLMI 2011. LNCS, vol. 7009, pp. 209–216. Springer, Heidelberg (2011)
17. Olivo-Marin, J.C.: Extraction of spots in biological images using multiscale products. Pattern Recognition **35**(9), 1989–1996 (2002)
18. Patricia, N., Caputo, B.: Learning to learn, from transfer learning to domain adaptation: a unifying perspective. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1442–1449. IEEE (2014)
19. Ribeiro, E., Shah, M.: Computer vision for nanoscale imaging. Machine Vision and Applications **17**(3), 147–162 (2006)

20. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011). pp. 833–840 (2011)
21. Tommasi, T., Orabona, F., Caputo, B.: Learning categories from few examples with multi model knowledge transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(5), 928–941 (2014)
22. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research **11**, 3371–3408 (2010)
23. Voss, N., Yoshioka, C., Radermacher, M., Potter, C., Carragher, B.: Dog picker and tiltpicker: software tools to facilitate particle selection in single particle electron microscopy. Journal of Structural Biology **166**(2), 205–213 (2009)
24. Woolford, D., Hankamer, B., Ericksson, G.: The laplacian of gaussian and arbitrary z-crossings approach applied to automated single particle reconstruction. Journal of Structural Biology **159**(1), 122–134 (2007)

Structures, Algorithms and Methods in Artificial Intelligence

BSO-FS: Bee Swarm Optimization for Feature Selection in Classification

Souhila Sadeg[1(✉)], Leila Hamdad[2], Karima Benatchba[3], and Zineb Habbas[4]

[1] Ecole nationale Supérieure d'Informatique, Oued Smar, Algiers, Algeria
s_sadeg@esi.dz
[2] LCSI, Ecole nationale Supérieure d'Informatique, Oued Smar, Algiers, Algeria
l_hamdad@esi.dz
[3] LMCS, Ecole nationale Supérieure d'Informatique, Oued Smar, Algiers, Algeria
k_benatchba@esi.dz
[4] LCOMS, Université de Lorraine, Metz, France
zineb.habbas@univ-lorraine.fr

Abstract. Feature selection is an important data-preprocessing step that often precedes the classification task. Because of large amount of features in real world applications, feature selection is considered as a hard optimization problem. For such problems, metaheuristics have been shown to be a very promising solving approach. In this work, we propose to use Bee Swarm Optimization (BSO) for feature selection. The proposed algorithm, BSO-FS, is based on the wrapper approach that uses BSO for the generation of feature subsets, and a classifier algorithm to evaluate the solutions. BSO-FS is tested on well-known datasets and its performances are compared with those of recently published methods. Obtained results show that for the majority of datasets, BSO-FS selects efficiently relevant features while improving the classification accuracy.

Keywords: Bee Swarm Optimization · Metaheuristic · Feature selection · Wrapper approach · Classification · Data mining

1 Introduction

Classification is a supervised learning task that is used in many real world problems. Its purpose is "to slot objects in a population into one or more categories based on a set of features measured on each object" [1]. To achieve this goal, a function (classifier) is inferred from labeled training dataset in order to correctly determine the class (category) of new examples.

In real-world application, especially when one is faced with unknown situations, many features are usually introduced in order to have the best possible representation without knowing, a priori, which of them are relevant. However, irrelevant or redundant features can lead to a classification accuracy decrease and to an unnecessary increase of computational cost [24][25]. Indeed, one might think that having more features would result in more discriminating power whereas, in practice, adding irrelevant or distracting attributes to a dataset often confuses

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 387–399, 2015.
DOI: 10.1007/978-3-319-19258-1_33

machine learning systems, although most of these algorithms are designed to learn which are the most appropriate attributes for making their decisions [2]. Therefore, selecting a subset of relevant features among a large initial set, often allows yielding better classification results. In addition, it offers other advantages such as facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and execution times, defying the curse of dimensionality to improve prediction performance [26].

Feature subset selection is usually considered as a data-preprocessing task that precedes a data mining one. It is defined by Kira and Rendell [31] as the problem of choosing a small subset of data attributes for a particular application that is, ideally, necessary and sufficient to describe the target. It can be achieved in two ways: One is to rank features according to some criterion and select the top k features. The other one is to select the smallest subset of features that does not deteriorate learning performance[3].

According to [30], a feature selection process, as illustrated in figure 1, mainly consists of four steps : subset generation, subset evaluation, stopping criterion, and result validation. The generation procedure is a search procedure that generates subsets of features. Each of them is evaluated according to the evaluation function in order to save the best one. This procedure is repeated iteratively until a stopping criterion is reached. It can be a predefined number of features, a predefined number of iterations, or some value of the evaluation function. Feature

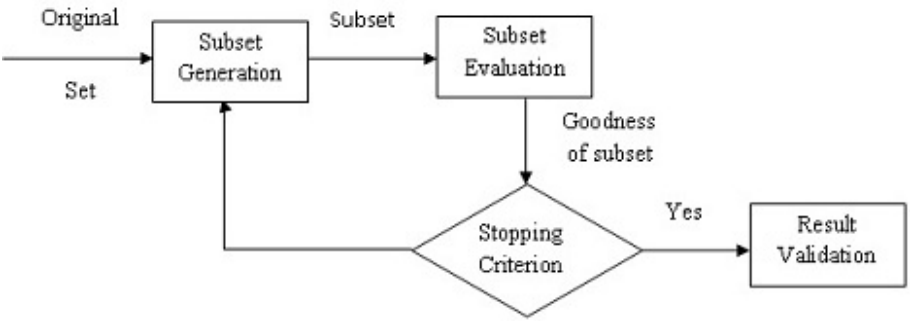


Fig. 1. Key steps for feature selection [30]

selection methods are generally divided into three categories according to how and when the relevance of the selected features is evaluated [4]. In other words, a feature selection method falls into one of the three categories depending on its use (or not) of a learning algorithm (a classifier) and on the way it uses it. In the filter approach, the algorithm selects the features by evaluating them while analyzing the general characteristics of data without involving any learning [4]. On the other hand, in the wrapper and embedded approaches, the feature selection method uses a learning algorithm but differ in the way they interact. Wheras

embedded approach integrates it in model building[3], the wrapper approach searches through the feature subset space and computes the estimated accuracy of a learning algorithm [8,11]. The feature space can be explored by using various strategies. Usually an exhaustive search is too expensive. Indeed, if the cardinality of the original set is N, finding the best subset requires to search among the 2^N candidate subsets which is computationally intractable [30]. Therefore, methods based on heuristics or random search are used to find good solutions in reasonable time following an evaluation function and a stopping criterion.

Among existing works on feature selection based on the wrapper approach and using metaheuristics, those using the genetic algorithm are the most frequent [6–9]. Metaheuristics based on Swarm Intelligence have also been applied to feature selection such as Ant Colony Optimization (ACO)[16,17,27,28], and Particle Swarm Optimization (PSO) [10,18].

In this paper, we propose a feature selection algorithm that follows the wrapper approach and uses the Bee Swarm Optimization algorithm (BSO) [12] as search algorithm. BSO is a swarm intelligence algorithm inspired from the foraging behavior of natural bees that has been successfully applied to various optimization problems [15,20–23]. Its good performances can be explained by a good balance between intensification and diversification, which leads respectively to a good exploitation and exploration of the search space. Here, we apply it for the first time to the feature selection problem with the aim to find the best feature subset that maximizes the classification accuracy. Experiments are conducted on a large number of data sets and the results are compared with those of PSO (Particle Swarm Optimization), ACO (Ant Colony Optimization), GA (Genetic Algorithm) and ABC (Artificial Bee Colony). The comparison showed that for most tested datasets, BSO outperformed the other methods.

The rest of the paper is organized as follows: In sections 2, the general algorithm of BSO is presented and its application for feature selection is described in section 3. Section 4 shows the results of BSO-FS and compares them to the results of other methods before we conclude with final remarks and directions for future works in section 5.

2 Bee Swarm Optimization Metaheuristic

Honeybee swarm in nature has a very interesting behavior characterized by collective intelligence that allows self-adaptation to the environment and dynamic tasks assignment. During the last decade, many new algorithms inspired by natural bees behavior were proposed and applied to different problems. A survey of these algorithms is given in [13].

Bee Swarm Optimization (BSO) metaheuristic proposed in [12] is inspired from the foraging behavior of real bees. Many researchers[5,14] have studied this behavior, considered as the most important task in the hive. In its foraging process, a bee starts by leaving the hive in order to find a flower and gather nectar. Then, it returns to the hive and unloads the nectar. If the food source is rich, the bee communicates its direction and distance to its mates via a dance. The other bees naturally follow the one that indicates the best food source.

BSO is an iterative search process based on a population of artificial bees cooperating to solve an instance of an optimization problem by imitating the foraging behavior described above. First, an initial solution is generated randomly or via a heuristic. This solution will be the reference solution, *refSol*, from which a search region, a set of candidate solutions, is determined. After that, each solution is assigned to a bee as a starting point of a local search. At the end of the search, each bee communicates its best found solution to the other ones through a table, named *dance*. One of the solutions in this table will be selected to be the reference solution in the next iteration. In order to avoid cycles, the reference solutions are stored in a Tabu list.

To ensure a good balance between exploitation and exploration, the selection of the reference solution follows the intensification and diversification principles. The first one aims to find good solutions by exploiting promising search region while the second allows a good coverage of the search space by visiting new regions. Intensification and diversification are performed according to the results obtained in a search region during a certain time: while the best global solution (bestGlobalSol) is improved, an intensification is performed by selecting a reference solution among those in dance table. Otherwise, either a diversification is performed, in case the search region has reached the maximum number of chances granted to it, or an intensification is performed and the number of chances is decreased. A diversification consists in choosing a solution from dance which is the furthest from all the solutions stored in the Tabu list. If all the solutions of dance are in Tabu list, a random solution is generated to be the next reference solution. The algorithm stops when the optimal solution is found or the maximum number of iterations is reached. The BSO general algorithm is described bellow.

3 BSO-FS: BSO for Feature Selection

In this paper, we propose a wrapper approach based method for feature selection that uses the BSO metaheuristic to perform the search process of subsets. Applying BSO to feature selection requires the adaptation of the general algorithm to the specificities of the problem. In this section, we will define how a solution is encoded, how its quality is evaluated, and how the search region is determined. The selection of the reference solution and the local search performed by the bees do not differ from the general algorithm given in the previous section.

- **Solution encoding**. A solution is represented by a binary vector of length n, where n is the original number of features. A position of the vector is set to 1 if the corresponding feature is selected and to 0 otherwise.
- **Fitness**. It represents the quality of the solution and is noted f(s). In our work, it is equivalent to the classification accuracy returned by the used classifier and is calculated by the following formula:

$$accuracy = \frac{number\,of\,true\,positive\ +\ number\,of\,true\,negative}{total\,population} \tag{1}$$

Algorithm 1. General algorithm BSO

```
Input: An instance of a combinatorial optimization problem
Output: The best solution found
refSol ← an initial solution found randomly or via a heuristic
while not stopping criterion do
    Insert refSol in Tabu list
    Determine searchRegion from refSol
    nbChances ←maxChances
    Assign a solution from searchRegion to each bee
    for each bee k do
        Perform a local search
        Store the result in the table dance
    if f(bestSol) > f(bestGlobalSol) then
        bestGlobalSol ←bestSol
        nbChances ←maxChances
        Intensification
    else
        if nbChances >0 then
            nbChances ←nbChances-1
            Intensification
        else
            diversification
return bestGlobalSol
```

Note that if two solutions have the same fitness, the one that uses less features is considrered to be the best.

– **Search region**. It is a set of solutions generated from the reference solution by flipping in refSol a number of bits equal to n/flip, flip being an empirical parameter. The size of this set equals the number of bees since each solution will be assigned to one bee as a starting point of its local search (see Figure 2).The value of flip has an impact on the performance of the research process because it determines the distance between refSol and the solutions which define the search region . Indeed, a too small value will favor the exploration instead of exploitation of the search space, while a too high value might lead the algorithm to converge to a local optimum.

To obtain the solutions of the search region we use two strategies that ensure that the obtained solutions are as distinct as possible. In the first one, the k^{th} solution is generated by flipping in refSol the variables separated by flip bits starting at the k^{th} variable. As an example, let n=20 and flip = 5. If the variables are subscripted from 0 to 19, then, as illustrated in figure 3, solutions s_0, s_1, s_2, s_3 and s_4 are obtained respectively by flipping the following bits: (0,5,10,15), (1,6,11,16), (2,7,12,17), (3,8,13,18) and (4,9,14,19). In the second strategy, a solution number k is obtained while flipping $n/flip$ contiguous bits starting by the k^{th} bit. Following the previous example the solutions s_0, s_1, s_2, s_3 and s_4

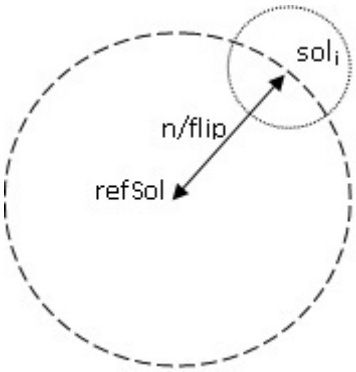


Fig. 2. Search region

are obtained respectively by flipping the following bits: (0,1,2,3) , (4,5,6,7) , (8,9,10,11) , (12,13,14,15) and (16,17,18,19).

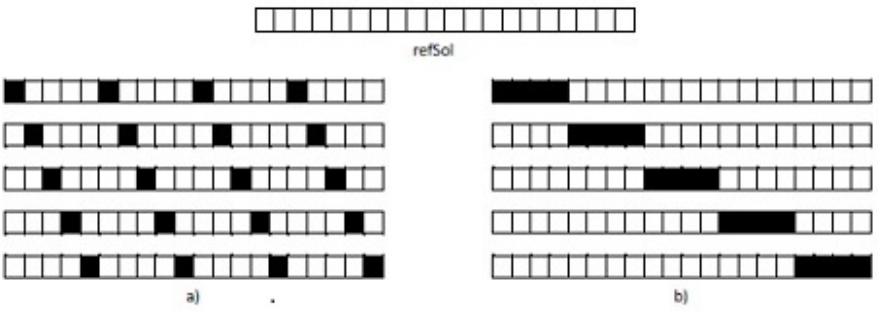


Fig. 3. a- solutions generated by the first strategy, b- solutions generated by the second strategy

4 Experimental Results

In order to investigate the performance of BSO-FS on feature selection, we implemented it in java programming language and used weka[1] and LibSVM[2] to execute the classification. Experiments were conducted on a personal computer running windows 7 System, and equipped with an Intel Core i3 2.20 GHz CPU and 6 GB RAM. In this section, we present the datasets used in the experimentations

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

then show the results obtained with BSO-FS. These results are compared to recent published results of GA, ACO, PSO and ABC metaheuristics. Only classification accuracies are compared since the algorithms are not run on similar machines.

4.1 Datasets

To evaluate our algorithm, we used Seventeen well-known benchmark classification datasets available in UCI Machine Learning Repository[3] that are frequently studied in machine learning. We considered datasets with considerable diversity in characteristics that are the number of features, classes and instances. According to the number of features, we considered small-sized datasets with less than 19 features (Iris, Breast Cancer, Glass, Diabetes, Heart-C, Heart-StatLog, Labor, Wine, Vehicle, Zoo, Labor), medium-sized datasets having between 20 and 49 features (German, Auto, WBCD, Ionosphere) and large sized datasets having more than 50 features (Lung cancer and Sonar). A summary of these datasets is shown in table 1.

Table 1. Datasets characteristics

Dataset	features	classes	instances	Dataset	features	classes	instances
Iris	4	2	150	Zoo	17	7	101
Breast-cancer	9	2	286	German	24	2	1000
Glass	9	7	214	WBCD	30	2	569
Diabetes	8	2	768	Wine	13	3	178
Heart-C	13	5	303	Ionosphere	34	2	351
Heart-StatLog	13	2	270	Vehicle	18	4	846
Hepatisis	19	2	155	Lung cancer	56	3	32
Labor	16	2	57	Sonar	60	2	208
Auto	25	7	205				

4.2 Results of BSO-FS

Like most metaheuristics, the performances of BSO-FS are greatly impacted by the values of its parameters. In our experiments, the optimal values were determined by performing a considerable number of runs using several values. Thus, the parameters were set to the following values: Flip = 5, MaxChances = 3 and Number of bees = 10 (or equal to the number of features if the latter is less than 10). Table 2 shows the accuracies obtained with BSO-FS using KNN classifier and applying 1 x 10 fold cross validation. For each dataset, we give the initial number of features, the accuracy obtained using all the features, the average classification accuracy over 10 runs, the average number of selected features, the best classification accuracy and the corresponding number of selected features.

[3] http://archive.ics.uci.edu/ml/

Table 2. Results of BSO using KNN classifier

Dataset	Features	Accuracy	Ave Accuracy	Ave Features	Best Accuracy	Features
Iris	4	95.33	96	2	96	2
Breast-cancer	9	72.38	76.57	5.3	76.57	4
Glass	9	70.56	79.44	5	79.44	5
Diabetes	8	70.18	71.48	3	71.48	3
Heart-C	13	76.24	83.50	6	83.50	6
Heart-StatLog	13	73.43	84.13	3	84.13	3
Hepatisis	19	80.64	90.97	7	90.97	7
Labor	16	82.46	98.25	8.8	98.25	7
Auto	25	76.1	87.80	6.3	87.80	5
Zoo	17	96.04	99.01	11.8	99.01	11
German	24	72	75.6	7	75.6	7
WBCD	30	95.96	98.24	14.3	98.24	11
Wine	13	94.94	99.44	7	99.44	7
Ionosphere	34	86.32	95.95	11.2	96.01	10
Vehicle	18	69.86	74.56	9.8	74.59	8
Lung-cancer	56	68.75	94.68	19.6	96.87	14
Sonar	60	86.54	98.22	27.3	98.56	23

The numerical results show that for all the benchmarks, the best accuracies are reached with reduced sets of features. For some datasets such as auto and lung cancer, the reduction rate is very significant. Also note that for almost all the benchmarks, the average results accuracies are equal to the best ones and this means that BSO-FS is a very stable algorithm thanks to the fact that it rarely uses randomness.

In order to analyze the behavior of BSO-FS, we perform 10 executions where the number of iterations equals 10. The fitness of the reference solutions are represented by a curve that illustrates the progression of the algorithm. The curves of 10 executions on lung-cancer dataset are represented in figure 4.

We can see that BSO-FS always reaches a very good solution (often its best solution) very quickly and maintains the same quality level until the end of the execution. The curves show some reference solution of lower quality that can be explained by a diversification performed by the algorithm.

4.3 Comparison with Other Algorithms

In order to further test the performances of BSO-FS, we compare our results with four recently published works based on the wrapper approach and using metaheuristics [10,18,19]. As these works do not use the same classifier, we carried out two series of tests using SVM and KNN classifiers. Table 3 compares the results of BSO-FS with those of PSO (Particle Swarm Optimization), ACO (Ant Colony Optimization), GA (Genetic Algorithm) and ABC (Artificial Bee Colony) using SVM [19]. The results show that BSO reaches better average accuracies than the others for 6 among the 9 datasets used in this part of the experiments. Indeed, ABC reached better results for Glass (71.50 %) and Labor

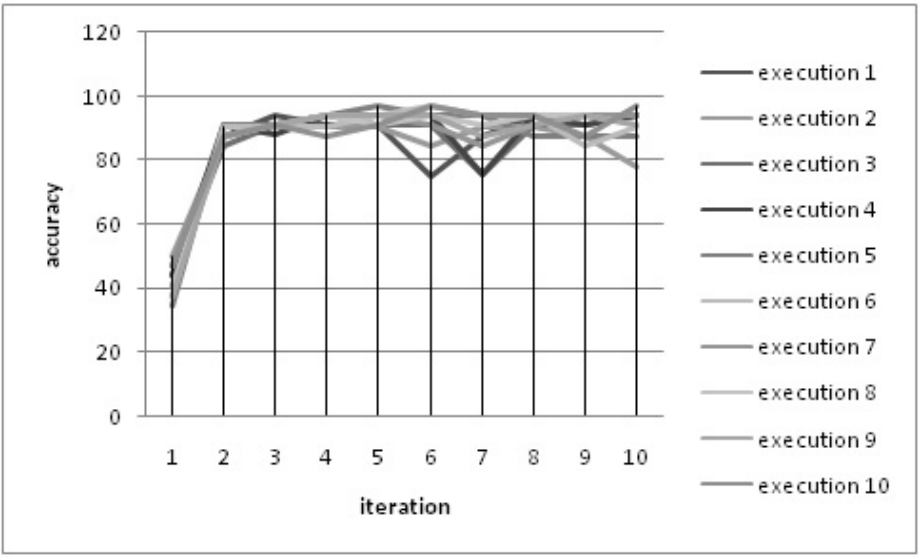


Fig. 4. Accuracies of ten solutions obtained on 10 executions on Lung-Cancer dataset

(98.26 %) for which BSO-FS obtained a very close result equal to 98.25 %. For the Diabetes dataset, we can observe that PSO, GA and ACO reached 75.65 % against 72.53 % for BSO-FS and 71.48 % for ABC. Nevertheless, we note that PSO, GA and ACO use all the features (8) whereas BSO-FS uses only 2 features and ABC only one feature. For the other datasets (Iris, Breast Cancer, Heart-C, Heart-StatLog, Hepatitis, Auto), BSO-FS obtained the best results. We can notice that the best accuracies are often obtained with the smallest subsets of features, the best example is the Auto dataset for which BSO-FS reached 86.34 % with only six features among the 25 constituting the original set.

In table 4, we present a comparison between our results and two variants of PSO recently published in literature [10,18]. It can be seen that the best classification accuracies are obtained with BSO-FS for all the benchmarks except the Vehicle dataset. We can also note that for the datasets that have the largest numbers such as Ionosphere, Sonar and Lung cancer, BSO-FS is much more efficient than PSO based methods.

5 Conclusion

In this paper, we proposed a wrapper approach based method for feature selection problem. Our method, named BSO-FS, uses the BSO metaheuristic to perform the search process in its subset generation step. The results show that for all datasets, reduced number of features allow to achieve better classification accuracy than the initial set of features. We also observed that, some equal size subsets could give different results, that means that some features are more

Table 3. Comparison with other algorithms using SVM classifier

Dataset	features	Algorithm	Ave features	Ave Accuracy
		BSO-FS	3	97.33
		PSO	4	96.66
Iris	4	ACO	4	96.66
		GA	4	96.66
		ABC	3	97.33
		BSO-FS	3.4	75.87
		PSO	8	73.08
Breast-cancer	9	ACO	9	73.08
		GA	8	73.08
		ABC	4	75.87
		BSO-FS	7	71.03
		PSO	8	71.03
Glass	9	ACO	8	71.03
		GA	8	71.03
		ABC	6	71.50
		BSO-FS	2	72.53
		PSO	8	75.65
Diabetes	8	ACO	8	75.65
		GA	8	75.65
		ABC	1	71.48
		BSO-FS	6	84.16
		PSO	8	83.17
Heart-C	13	ACO	7	80.86
		GA	7	80.20
		ABC	7	83.17
		BSO-FS	3.4	84.87
		PSO	8	82.96
Heart-Statlog	13	ACO	7	81.11
		GA	8	73.70
		ABC	3	84.81
		BSO-FS	8.7	87.74
		PSO	7	86.45
Hepatitis	19	ACO	7	83.23
		GA	7	83.26
		ABC	9	87.10
		BSO-FS	7.8	98.25
		PSO	5	89.47
Labor	16	ACO	6	92.98
		GA	9	89.47
		ABC	8	98.26
		BSO-FS	6	86.34
		PSO	8	68.78
Auto	25	ACO	9	72.20
		GA	9	69.21
		ABC	9	82.93

Table 4. Comparison with other algorithms using KNN classifier

Dataset	features	Algorithm	Ave features	Ave Accuracy	best accuracy	standard deviation
Zoo	17	BSO-FS	11.8	99.01	99.01	0.00
		PSO [10]	6.46	95.52	97.14	1.62
German	24	BSO-FS	7	75;6	75.6	0.00
		PSO [10]	12.76	68.47	70.67	2.20
WBCD	30	BSO-FS	14.3	98.24	98.24	0.00
		PSO [18]	14.2	98.17	97.24	0.07
		PSO [10]	3.46	93.98	94.74	0.76
Wine	13	BSO-FS	7	99.44	99.44	0.00
		PSO [18]	8	99.44	99.44	0.00
		PSO [10]	6.84	95.26	98.77	3.51
Ionosphere	34	BSO-FS	11.2	95.95	96.01	0.06
		PSO [10]	3.26	87.27	91.43	4.16
Vehicle	13	BSO-FS	7	99.44	99.44	0.00
		PSO [18]	8	99.44	99.44	0.00
		PSO [10]	6.84	95.26	98.77	3.51
Lung-Cancer	56	BSO-FS	19.6	94.68	96.87	2.19
		PSO [10]	6.74	78.4	90	11.60
Sonar	13	BSO-FS	27.3	98.22	98.56	0.34
		PSO [18]	30.2	96.92	97.12	0.20
		PSO [10]	11.24	78.16	85.71	7.55

relevant when they are associated with others. The study shows that BSO-FS is an effective search algorithm. Indeed, It gives very promising results after few iterations of the search process. The results show besides that BSO-FS is highly stable since it generally gives the same results at each execution, which is translated by a very small standard deviation often equal to zero. This is explained by a good balance between exploitation and exploration ensured by a good compromise between intensification and diversification. The application of these fundamental principles of BSO is allowed by a right setting of the Flip and MaxChances parameters. Finally, the comparison of the results of BSO-FS with the results of other recently published metaheuristics shows that our algorithm gives generally best accuracies with good reduction rates and very little deviations. In our future works, we plan to investigate hybridization of BSO-FS with filter algorithms in the phase of initialization and to parallelize the search process of the bees in order to perform experiments on larger datasets.

References

1. Clarke, B., Fokoue, E., Zhang, H.H.: Principles and theory for data mining and machine learning. Springer Science Business Media (2009)
2. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2005)
3. Liu, H., Motoda, H. (eds.).: Computational methods of feature selection. CRC Press (2007)
4. Liu, H., Motoda, H., Setiono, R., Zhao, Z.: Feature Selection: An Ever Evolving Frontier in Data Mining. FSDM **10**, 4–13 (2010)

5. Von Frisch, K., Lindauer, M.: The "Language" and Orientation of the Honey Bee. Annual Review of Entomology **1**, 45–58 (1956, 1973)
6. Oliveira, L.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. International Journal of Pattern Recognition and Artificial Intelligence **17**(06), 903–929 (2003)
7. Inza, I., Merino, M., Larraaga, P., Quiroga, J., Sierra, B., Girala, M.: Feature subset selection by genetic algorithms and estimation of distribution algorithms: a case study in the survival of cirrhotic patients treated with TIPS. Artificial Intelligence in Medicine **23**(2), 187–205 (2001)
8. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. In: Feature Extraction, Construction and Selection, pp. 117–136. Springer, US (1998)
9. Huang, J., Cai, Y., Xu, X.: A hybrid genetic algorithm for feature selection wrapper based on mutual information. Pattern Recognition Letters **28**(13), 1825–1844 (2007)
10. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing **18**, 261–276 (2014)
11. Yusta, S.C.: Different metaheuristic strategies to solve the feature selection problem. Pattern Recognition Letters **30**(5), 525–534 (2009)
12. Drias, H., Sadeg, S., Yahi, S.: Cooperative bees swarm for solving the maximum weighted satisfiability problem. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 318–325. Springer, Heidelberg (2005)
13. Karaboga, D., Akay, B.: A survey: algorithms simulating bee swarm intelligence. Artificial Intelligence Review **31**(1–4), 61–85 (2009)
14. Seeley, T.D.: Honeybee ecology: a study of adaptation in social life. Princeton University Press (2014)
15. Sadeg, S., Drias, H.: A selective approach to parallelise Bees Swarm Optimisation metaheuristic: application to MAX-W-SAT. International Journal of Innovative Computing and Applications **1**(2), 146–158 (2007)
16. Kabir, M.M., Shahjahan, M., Murase, K.: A new hybrid ant colony optimization algorithm for feature selection. Expert Systems with Applications **39**(3), 3747–3763 (2012)
17. Robbins, K.R., Zhang, W., Bertrand, J.K., Rekaya, R.: The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. Mathematical Medicine and Biology **24**(4), 413–426 (2007). ISO 690
18. Chuang, L.Y., Tsai, S.W., Yang, C.H.: Improved binary particle swarm optimization using catfish effect for feature selection. Expert Systems with Applications **38**(10), 12699–12707 (2011)
19. Schiezaro, M., Pedrini, H.: Data feature selection based on Artificial Bee Colony algorithm. EURASIP Journal on Image and Video Processing **2013**(1), 1–8 (2013)
20. Belkebir, R., Guessoum, A.: A hybrid BSO-Chi2-SVM approach to Arabic text categorization. In: 2013 ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1–7. IEEE, May 2013
21. Drias, H., Mosteghanemi, H.: Bees swarm optimization based approach for web information retrieval. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 6–13. IEEE, August 2010

22. Djeffal, M., Drias, H.: Multilevel bee swarm optimization for large satisfiability problem instances. In: Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., Yao, X. (eds.) IDEAL 2013. LNCS, vol. 8206, pp. 594–602. Springer, Heidelberg (2013)
23. Djenouri, Y., Drias, H., Chemchem, A.: A hybrid bees swarm optimization and tabu search algorithm for association rule mining. In: 2013 World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 120–125. IEEE, August 2013
24. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial intelligence **97**(1), 245–271 (1997)
25. Koller, D., Sahami, M.: Toward optimal feature selection (1996)
26. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research **3**, 1157–1182 (2003)
27. Alsukker, A., Khushaba, R., Al-Ani, A., Al-Jumaily, A.A.: Enhanced feature selection algorithm using ant colony optimization and fuzzy memberships. IASTED (2008)
28. Ke, L., Feng, Z., Ren, Z.: An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recognition Letters **29**(9), 1351–1357 (2008)
29. Ke, L., Feng, Z., Ren, Z.: An efficient ant colony optimization approach to attribute reduction in rough set theory. Pattern Recognition Letters **29**(9), 1351–1357 (2008)
30. Dash, M., Liu, H.: Feature selection for classification. Intelligent data analysis **1**(3), 131–156 (1997)
31. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: AAAI, vol. 2, pp. 129–134, July 1992

Improved Retrieval for Challenging Scenarios in Clique-Based Neural Networks

Xiaoran Jiang(✉), Max Raphael Sobroza Marques, Pierre-Julien Kirsch, and Claude Berrou

Télécom Bretagne, Electronics department, UMR CNRS Labsticc, Technopôle Brest Iroise-CS 83818, 29238 Brest Cedex, France
xiaoran.jiang@telecom-bretagne.eu

Abstract. This paper describes new retrieval algorithms based on heuristic approach in clique-based neural networks introduced by Gripon and Berrou. This associative memory model resembles the well-known Willshaw model with specificity of clustered structure. Several retrieval algorithms exist, for instance, Winners-Take-All and Losers-Kicked-Out. These methods work generally well when the input message suffers reasonable distortions, but the performance drops dramatically in some challenging scenarios because of severe interference. By means of simulations, we show that the proposed heuristic retrieval algorithms are able to significantly mitigate this issue while maintaining biological plausibility to some extent.

Keywords: Associative memory · Retrieval algorithms · Sparse coding · Recurrent neural networks

1 Introduction

Unlike traditional computer memory, such as random access memory (RAM) where data are returned by using memory addresses, associative memory (also called content-addressable memory (CAM)) uses a different strategy: the user supplies a data word and the associative memory searches its entire memory to see if that data word is stored anywhere in it. Some neuro-inspired associative memories are devices that mimic the brain memory in some aspects: they can store messages and retrieve them in presence of distortions, for instance, errors or erasures; if no word is found in the memory, this will return an associated piece of data; instead of a brute-force approach, the retrieval process should maintain low complexity and be energy-efficient.

Among popular neuro-inspired associative memories, we can cite Hopfield neural networks (HNN) [1,2] and Willshaw networks [3,4]. The former uses

This work was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement number 290901.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 400–414, 2015.
DOI: 10.1007/978-3-319-19258-1_34

binary threshold nodes and stores messages that are as long as the size of the network via a weighted connection updating rule. The latter exploits the interest of sparse coding by addressing a message to only a small number of nodes in the network. For a detailed recent state-of-art of associative memories, see [5] and the references therein.

By introducing an organization in clusters, Gripon and Berrou recently proposed a Willshaw-type neural network based on cliques and sparse representations [6] that can treat information with non binary alphabets. We call this network GBNN in the sequel. This is extended in [7] with a further sparse organization and in [8,9], authors adapt GBNN to store long sequences by incorporating oriented links. These models demonstrate large storage diversity (the number of storable messages with a relatively high recovery rate) and capacity (the amount of storable information), as well as strong robustness with respect to erasures and errors.

Retrieving a message in GBNN and its variants (we collectively name them as "clique-based neural networks") is similar to maximal clique (a clique that cannot be enlarged) retrieval problem. As it is well known, the problem of listing all maximal cliques is NP-complete: it may require exponential time as there exist graphs with exponentially many maximal cliques. However, in neuro-inspired models, retrieval algorithms are not for the purpose of listing all maximal cliques with zero error rate, but to give a likely answer according to certain distance metrics while maintaining polynomial complexity as well as a certain degree of biological plausibility. Several retrieval algorithms exist in GBNN and its variants, such as Winner-Takes-All (WTA) and Losers-Kicked-Out (LsKO) [10]. These methods work generally well when the input message suffers reasonable distortions, but performance drops dramatically in some challenging scenarios because of severe interference. In this paper, we intend to propose some new heuristic retrieval algorithms in order to mitigate this issue.

The rest of this paper is organized as follows: Section 2 recalls the definition of a message and how to map and store messages in clique-based neural networks. Section 3 explains classical retrieval algorithms in such networks. New retrieval algorithms with heuristic process are proposed in Section 4. Section 5 assesses retrieval performance with synthetic data. Finally, a conclusion is drawn in Section 6.

2 Clique-Based Neural Networks

2.1 Message

Any natural object can be defined by a set of elementary features. For instance, the object *"car"* can be described by the relation

$$\text{CAR} \leftarrow \{\text{four wheels, engine, machine, road, Carl Benz}\}$$

and the object *"computer"* by another relation

$$\text{COMPUTER} \leftarrow \{\text{binary, intelligence, machine, calculation, Turing}\}.$$

Objects may share some common features: both *car* and *computer* are *machines*, but other features are sufficient to distinguish the two. A relation is said unambiguous if a unique object is addressed by this relation.

In order to adapt to the information theory context and to develop this paper on a more abstract level, we call the above relations *messages* and the features *segments*. A well designed associative memory mimicking brain behavior should be able to retrieve the original message from its distorted version. Several forms of distortions are envisaged. this cluster are activated; an error, the situation when the fanal activated in a cluster is not the right one; and an insertion, the situation when some fanals are activated in the clusters that should be silent.

1. *erasure*: Some segments of the message are omitted, e.g. {four wheels, engine, ? , ?, Carl Benz}.
2. *insertion*: Besides the correct segments, some segments that do not belong to the message are also inserted, e.g. {four wheels, engine, machine, road, Carl Benz, apple, neuron}.
3. *error*: In the place of the correct segments, some incorrect ones are provided, e.g. {four wheels, engine, animal, ocean, Carl Benz}.
4. *blur*: One is not sure of the exact value of some segments, e.g. {four wheels, engine, machine, road, (Carl Benz, Carl Zeiss, Carl Lewis)}.

2.2 Message Storing

Let us now recall the structure of clique-based neural networks initially introduced in [6] and then extended and studied in depth in [7,8]. This network could be seen as a Willshaw structure [3,4] with the ability to store messages with a non-binary alphabet.

Such a network is composed of n binary nodes organized in χ clusters. Each cluster contains a certain number of nodes, which materialize the alphabet elements. We denote by n_{ij} or equivalently the 2-tuple (i, j) the j^{th} node in the i^{th} cluster. Different clusters may reify different alphabets. For instance, there may be a cluster dedicated to fruits and another one to condiments. A node in cluster "Fruit" may then represent "apple", "peach" or "orange", etc. In a similar way, a node in cluster "Condiment" may represent "salt", "sugar" or "pepper", etc. Now let us consider the message

$$\text{APPLE PIE} \leftarrow \{\text{apple, cream, pastry, sugar, American}\}.$$

Graphically, each segment of the message APPLE PIE is mapped to a unique node of the corresponding cluster. As a consequence, storing a message is equivalent to storing the corresponding pattern of c nodes, c being the number of segments that compose the message. Then, these selected nodes are fully interconnected between them to form the clique APPLE PIE.

It is important to note that all the connections in this network are binary. That means, if the clique APPLE PIE has been already stored, the storage of another clique ICE CREAM will not increment the weight of the connection

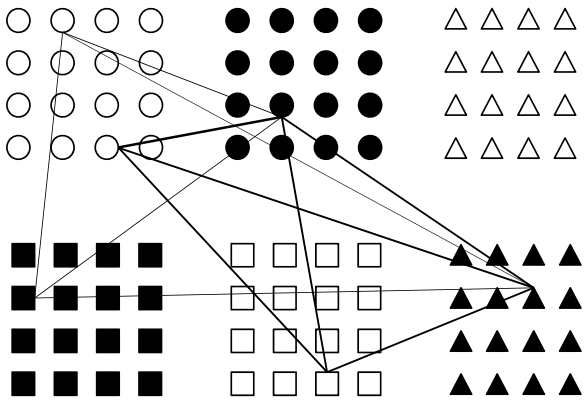


Fig. 1. Storing process illustration for clique-based networks with a sparse structure. Two messages of order $c = 4$ are stored in a network of $\chi = 6$ clusters of $l = 16$ fanals each in form of cliques. One edge is shared. Fanals are represented respectively in 6 clusters by open circles, filled circles, open triangles, filled rectangles, open rectangles and filled triangles.

linking nodes "cream" and "sugar", although the latter also contains these two segments.

Let us denote by $w_{(i,j)(i',j')}$ the weight between nodes (i, j) and (i', j'). Formally, the storage process can be summarized in the form of the weight equation:

$$w_{(i,j)(i',j')} = \begin{cases} 1, & \text{if } i \neq i' \text{ and } \exists \mathbf{m} \in \mathcal{M}, m_i = j, m_{i'} = j'; \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

with m_i being segment i of message $\mathbf{m}$ and $\mathcal{M}$ the entire set of stored messages.

Two network structures exist. In a GBNN structure, all the clusters are addressed by any single message, whereas in a sparse structure, only a small part of the network is addressed at a time for information acquisition. The latter is represented in Fig. 1.

It is worthy to note that the high memory efficiency offered by clique-based networks to store messages is due to binary connections. This has to be looked at in strong contrast with conventional neural networks based on analog or analog-like links, such as Hopfield-type networks and Boltzmann machines [11]. Compared to Willshaw-type networks where the connections are also binary, the mapping of clique-based networks with the external world is much more natural since clusters may materialize non binary alphabets.

3 Retrieval Algorithms

As said in Section 2.1, an associative memory should be able to retrieve the original message $\mathbf{m}$ from its distorted version $\hat{\mathbf{m}}$. In clique-based networks, relying on the set of existing connections, an appropriate decoding algorithm aims

to give an estimate $\tilde{\mathbf{m}}$, which would be the nearest from $\hat{\mathbf{m}}$ in terms of Hamming distance among $\mathcal{M}$: minimum Hamming distance amounts to maximum correspondence between two graphical patterns.

The stimuli corresponding to $\hat{\mathbf{m}}$ activate nodes with a score of 1:

$$\forall i, j, v(n_{ij}) \leftarrow \begin{cases} 1, & \text{if } n_{ij} \in \hat{\mathbf{m}}_i; \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

with $v(n_{ij})$ being the score of node n_{ij}. The retrieval algorithms could be then iterative and each iteration generally consists of two steps: *message passing* and *selection of winners*. The message passing propagates stimuli (the provided segments $\hat{\mathbf{m}}_i$) within the whole network via established connections and the contributions are added at each node. Formally, it can be expressed by Eq. (3):

$$\forall i, j, v(n_{ij}) \leftarrow \sum_{i'=1}^{\chi} \max_{1 \le j' \le l} \left[w_{(i',j')(i,j)} v(n_{i'j'}) \right] + \gamma v(n_{ij}) \tag{3}$$

with γ being the memory effect, which we usually set to 1. There are also other alternatives of message passing algorithms, and a more detailed study of them has been presented in [10]. We use Eq. (3) in this paper because it is universal and offers best results for both GBNN and sparse networks. We now concentrate on the selection of winners algorithms that follow the message passing step. This step aims to choose the nodes that are the most likely for the next iteration.

3.1 Local Winner-Takes-All (LWTA)

In the case that all the clusters are used to store a message (i.e. GBNN), a simple Winner-Takes-All rule can be performed locally within each cluster.

$$\forall j \in [|1, l|], v(n_{ij}) \leftarrow \begin{cases} 1 & \text{if } v(n_{ij}) = v_i^{\max} \text{ and } v_i^{\max} \ge \sigma; \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $v_i^{\max}$ is the largest score in cluster i after the message passing step and σ a predefined threshold. After these operations, all nodes have value 1 or 0. This is quite comparable to the neurons of the very simple McCulloch-Pitts model [12] with binary inputs and outputs, in which the only functionality of a neuron is the summation of inputs and the comparison with some firing threshold. Note that this remark is also valid for the following rules.

3.2 Global Winners-Take-All (GWsTA)

The Local Winner-Takes-All can be adapted in the sparse structure to become Global Winners-Take-All. Instead of choosing one node that has the maximal score per cluster, the algorithm chooses at least s nodes that have the maximal score or near maximal scores. If several nodes have the same score as the s^{th} does, these nodes are activated as well. We also call it "$s-$GWsTA". Ideally, the

parameter s should be set to c, the order of the message to recover. Nevertheless, the retrieval performance is relatively flexible to the choice of value of s. This algorithm is quite comparable to the "Strategy Constant Activity" [13] applied in Willshaw networks. A simple version of s-GWsTA is proposed in Algorithm 1.

```
begin
    μ ← 0, v ← v^max, m̃ ← ∅;
    while μ < s do
        for 1 ≤ i ≤ n do
            if v(n_ij) = v then
                add n_ij to m̃;
                μ ← μ + 1;
            end
        end
        v ← v − 1;
    end
    Return m̃;
end
```

Algorithm 1. A simple implementation of the s-GWsTA algorithm

3.3 Losers-Kicked-Out (LsKO)

The algorithm of Losers-Kicked-Out could be regarded as the opposite of GWsTA to some extent. Nevertheless, in contrast to other algorithms, LsKO proposes to work in a limited subgraph of the network, that is, the subgraph carried by the estimate $\tilde{\mathbf{m}}(t-1)$, t being the number of the current iteration. If all the nodes in $\tilde{\mathbf{m}}(t-1)$ do not have the same score, those having the minimal score are eliminated:

$$(v_{\max}, v_{\min}) \leftarrow \left(\max_{n_{ij} \in \tilde{\mathbf{m}}(t-1)} v(n_{ij}), \min_{n_{ij} \in \tilde{\mathbf{m}}(t-1)} v(n_{ij}) \right); \tag{5}$$

if $v_{\max} \neq v_{\min}$,

$$\tilde{\mathbf{m}}(t) \leftarrow \{n_{ij} \mid n_{ij} \in \tilde{\mathbf{m}}(t-1), v(n_{ij}) > v_{\min}\}. \tag{6}$$

This process can be iterative. If the stopping condition is not satisfied, that is, all the remaining fanals do not have the same score, more iterations of LsKO should be applied.

4 Heuristic Retrieval by Boosting in Challenging Scenarios

The algorithms described in Section 3 perform generally well when the input message suffers reasonable distortions. LWTA and GWsTA are efficient and converge faster when the distortion of the initial pattern is small, whereas LsKO

retrieves better when there are insertions and errors. However, their performance drops dramatically in some challenging scenarios because of severe interference. Fig. 2 describes a decoding scenario in a sparse network. Each node belongs to its own cluster, and for the reason of clarity, clusters are omitted. The target message to recover is illustrated by four filled circles, that is, the clique {A, B, C, D}. The other 8 nodes represented by empty circles are insertions (i.e. false stimuli). The scores of all nodes are calculated according to Eq. (3) with $\gamma = 1$. In this particular case, if we only consider the first iteration of algorithms, none of the retrieval algorithms described in Section 3 will succeed in recovering the correct clique: both GWsTA and LsKO would eliminate nodes A and D.

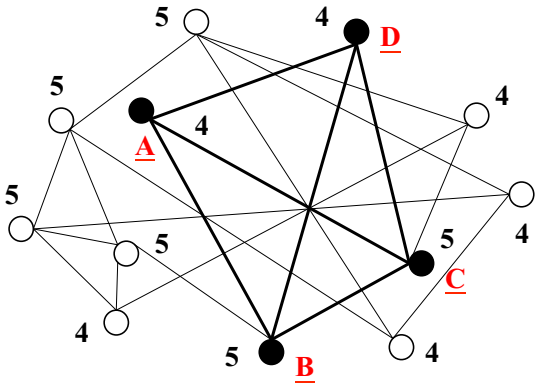


Fig. 2. Illustration of a challenging decoding scenario in clique-based networks in which both GWsTA and LsKO fail to retrieve correctly the original message {A, B, C, D}

One should distinguish two types of errors: the structural errors that are caused by incremental density of the network (in an extreme case, a fully saturated network with density 100% would lose capability to retrieve any message); and those generated because of the imperfection of retrieval algorithms. In Fig. 2, the decoding issue is caused by the latter. In fact, the clique {A, B, C, D} is the only clique of order 4 and moreover is the maximal clique in this graph.

In this section, we propose new algorithms that are able to overcome this issue for such challenging scenarios. In the scope of this paper, we focus on two scenarios: 1) large number of insertions in sparse networks; and 2) significant blurs in GBNN networks. The former implies that at the input, large quantity of stimuli would be present in other clusters than those embedding the target clique, whereas the latter implies numerous competitors within the same cluster. In both scenarios, we suppose that the original message is also fully provided.

These new algorithms carry out a heuristic process. Some node in the input graph will be boosted by a strong impulsive activity and then this boosted activity will propagate towards its adjacent nodes. This process allows extracting

a subgraph of a much smaller order within which classical retrievals will be thereafter performed. The algorithm changes the node to boost before reaching a certain stopping condition. More detailed explanations are provided in the following two subsections.

4.1 Retrieval by Boosting in Sparse Networks

Let us consider a sparse network storing messages via the adjacency matrix W. A heavily distorted input message $\hat{\mathbf{m}}$ is provided to the decoder and the number of insertions c_{ins} is much larger than the message order c, $c_{\text{ins}} \gg c$. An implementation of heuristic retrieval by boosting is described in Algorithm 2.

At the beginning, an optional pruning step (Lines 3-8) could be performed in order to eliminate the nodes with too weak connectivity, which probably do not take part of the right clique. The scores are calculated via the message passing equation, and all those nodes having a score smaller than a predefined threshold σ_p are eliminated. The resulting intermediate pattern after this step is denoted by $\tilde{\mathbf{m}}'$. Then, a node in $\tilde{\mathbf{m}}'$ is randomly selected and boosted by an impulsive activity g (Line 13). A message passing step (Lines 14-19) is performed to determine the subgraph that is exclusively composed of its direct neighbors: all the nodes having a score smaller than g are eliminated. Then, instead of performing LsKO or GWsTA within the whole graph, classical retrieval process is applied only to this subgraph of a much smaller order. A candidate clique is extracted at the end of this process (Line 20). One should then verify (Lines 21-23) if the clique order surpasses a predefined target order c', which can be reflected by the score of nodes (the scores will be c for all nodes of a clique of order c if convergence is reached). The value of c' should ideally be set to c, the order of the target clique, if the optional pruning step is not previously performed. Otherwise, we use a slightly smaller value of c'. Random boosting will be re-processed if the previous candidate clique does not satisfy this order criterion. The algorithm ends if the order criterion is satisfied or a certain limit of boosting attempts $t_{\max}$ is reached.

4.2 Retrieval by Boosting in GBNN Networks

Let us now consider a network storing messages that address all the clusters. Algorithm 2 designed for sparse networks can also be applied to this structure. Furthermore, it is to our advantage to adapt this algorithm into a more effective version thanks to interesting properties of GBNN.

First, in GBNN networks, it is known [14] that if the target pattern is provided among the input pattern, it will remain active during all the iterations. As a result, the preliminary pruning step becomes sure and efficient by simply applying several iterations of LWTA. Second, since the target clique contains exactly one node per cluster, the heuristic boosting process could be applied only to the cluster that contains the largest number of activated nodes. That reduces significantly the number of attempts with reasonable loss in error rate. Third,

the order criterion could be replaced: the algorithm ends if no more than one node per cluster remains active. An implementation of retrieval by the boosting algorithm in GBNN networks is detailed in Algorithm 3.

Input: $\hat{\mathbf{m}}$, W, pruning threshold σ_p (optional), retrieval order threshold c', limit number of boosting attempts $t_{\max}$
Output: retrieved message $\tilde{\mathbf{m}}$

begin
  /* initialisation */
  $\tilde{\mathbf{m}}' \leftarrow \hat{\mathbf{m}}$;
  /* Optional step of pruning, eliminate the nodes that have too weak connectivity */
  for $1 \leq i \leq$ **Order**$(\hat{\mathbf{m}})$ **do**
    calcul $v(\hat{m}_i)$ using W according to Eq. (3);
    if $v(\hat{m}_i) < \sigma_p$ **then**
      remove $\hat{m}_i$ from $\tilde{\mathbf{m}}'$
    end
  end
  $t \leftarrow 0$;
  $\tilde{m} \leftarrow \emptyset$;
  while **Order**$(\tilde{\mathbf{m}}) < c'$ *and* $t < t_{max}$ **do**
    generate a random integer value j between 1 and **order**$(\tilde{\mathbf{m}}')$;
    /* Random boosting */
    $v(\tilde{m'}_j) \leftarrow g$, g being an integer larger than **Order**$(\hat{\mathbf{m}})$;
    for $1 \leq i \leq$ **Order**$(\tilde{\mathbf{m}}')$ **do**
      calcul $v(\tilde{m'}_i)$ using W according to Eq. (3);
      if $v(\tilde{m'}_i) \geq g$ **then**
        add $\tilde{m'}_i$ to $\tilde{\mathbf{m}}$
      end
    end
    perform iterations of LsKO or GWsTA to $\tilde{\mathbf{m}}$ until convergence, the result replaces $\tilde{\mathbf{m}}$;
    if **Order**$(\tilde{\mathbf{m}}) < c'$ **then**
      $\tilde{m} \leftarrow \emptyset$
    end
    $t \leftarrow t + 1$;
  end
  /* Optional step */
  perform iterations of GWsTA to $\tilde{\mathbf{m}}$ until convergence, the result replaces $\tilde{\mathbf{m}}$;
  Return $\tilde{\mathbf{m}}$;
end

Algorithm 2. An implementation of retrieval by the boosting algorithm in sparse networks

Input: input message $\hat{\mathbf{m}}$, adjacency matrix W
Output: retrieved message $\tilde{\mathbf{m}}$

begin
  /* Initialisation */
  $\tilde{m} \leftarrow \emptyset$;
  /* Eliminate interfering nodes */
  perform iterations of LWTA or simple thresholding ($\sigma = \chi$) to $\hat{\mathbf{m}}$, the result updates $\hat{\mathbf{m}}$;
  /* Boosting in the cluster that has the largest number of activated nodes */
  choose the segment $\hat{m}_k$ of $\hat{\mathbf{m}}$ that contains the largest number of nodes: $\mathbf{card}(\hat{m}_k) = \max_j \mathbf{card}(\hat{m}_j)$;
  for $1 \leq i \leq \mathbf{card}(\hat{m}_k)$ **do**
    let n_k^i be the i^{th} node activated in segment k;
    $v(n_k^i) \leftarrow g$, g being an integer larger than χ, the number of clusters;
    for $1 \leq k' \leq \chi$ **do**
      for $1 \leq i' \leq \mathbf{card}(\hat{m}_{k'})$ **do**
        calcul $v(n_{k'}^{i'})$ using W according to Eq. (3);
        if $v(n_{k'}^{i'}) \geq g$ **then**
          add $n_{k'}^{i'}$ to $\tilde{\mathbf{m}}$;
        end
      end
    end
    perform iterations of LWTA or simple thresholding ($\sigma = \chi$) to $\tilde{\mathbf{m}}$ until convergence, the result replaces $\tilde{\mathbf{m}}$;
    if *there are more than 1 node per segment in* $\tilde{\mathbf{m}}$ **then**
      $\tilde{\mathbf{m}} \leftarrow \emptyset$
    end
    else
      exit the loop;
    end
  end
  Return $\tilde{\mathbf{m}}$;
end

Algorithm 3. An implementation of retrieval by the boosting algorithm in GBNN networks for blurred messages

5 Performance Evaluation

5.1 Blurred Messages

A message is considered as blurred if we are not sure of some of its segments. An example of blurred messages was given in Section 2.1: {four wheels, engine, machine, road, (Carl Benz, Carl Zeiss, Carl Lewis)}. The name of the inventor is ignored, but somehow one roughly recalls that the name is started with "Carl". In this case, ambiguities exist among three names: Carl Benz, Carl Zeiss and Carl Lewis. Back to our network topology, this is reflected by the simultaneous

activation of three nodes in their dedicated cluster. This number is called the span of blur and is denoted by b. The larger the value of b, the more numerous blurred clusters, more difficult will be the retrieval process.

Let us suppose that there are M messages of order χ stored in a GBNN network. Each cluster contains l nodes. Suppose that each segment of the input message is blurred with span b. Thus, $b\chi$ nodes (among them there may exist b^χ potential cliques) are activated at the beginning of the retrieval process. Among them, only one node per cluster is correct. The retrieval error probability after one iteration of LWTA has been proven in [15]:

$$P_e \approx 1 - \left[1 - \left(1 - (1-d)^b\right)^{\chi-1}\right]^{\chi(b-1)} \tag{7}$$

with density d equal to:

$$d = 1 - \left(1 - \frac{1}{l^2}\right)^M.$$

If several iterations are performed, the retrieval performance can be significantly better than that expressed by Eq.(7)[15]. However, a large margin of improvement is remaining. Indeed, a well designed algorithm should aim to find the only clique among the b^χ possible ones, and the retrieval error rate should be lower bounded by the probability of having another clique among the initially activated pattern. Let us estimate this lower bound.

Suppose there is an interfering clique with i vertices that differ from the target clique. The value of i can be between 1 and χ. Besides the connections that this clique shares with the target one, the number of spurious connections is

$$(\chi - i) \times i + \frac{i \times (i-1)}{2}.$$

And there could be $\binom{\chi}{i} \times (b-1)^i$ such cliques. If we assume the independence of connections[1], the probability of not having an interfering clique with i vertices that differ from the target clique is:

$$\left(1 - d^{(\chi-i)\times i + \frac{i\times(i-1)}{2}}\right)^{\binom{\chi}{i}\times(b-1)^i}.$$

This should be verified for all i between 1 and χ. Thus, the theoretical lower bound P_e^b of error probability P_e can be expressed in Eq. (8):

$$P_e^b = 1 - \prod_{i=1}^{\chi} \left(1 - d^{(\chi-i)\times i + \frac{i\times(i-1)}{2}}\right)^{\binom{\chi}{i}\times(b-1)^i}. \tag{8}$$

One can easily verify that the error rate described in Eq. (7) is largely higher than the lower bound in Eq. (8).

[1] We have to admit that this hypothesis is true for Erdős-Rényi models where connections are randomly and independently generated, but less true for clique-based networks.

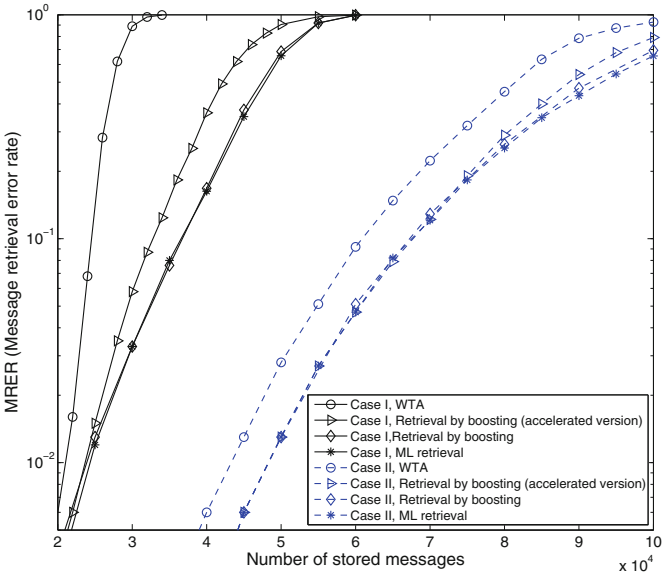


Fig. 3. Performance comparison of LWTA and retrieval by boosting in GBNN for blurred messages. Two cases are simulated: Case I: $\chi = 8, l = 256, b = 9$; Case II: $\chi = 12, l = 256, b = 3$. The curves of ML decoding are also provided as references.

In Fig. 3, we assess the effectiveness of boosting algorithms to mitigate this gap in the case of blurred messages. Messages stored in GBNN networks are randomly generated. Two cases are simulated: Case I: $\chi = 8, l = 256, b = 9$; Case II: $\chi = 12, l = 256, b = 3$. The curves of Maximum Likelihood (ML) decoding are also provided as references, as ML is the best algorithm in terms of error rate in any storing system. ML decoding is implemented as a brute-force search to list all the cliques contained in the input blurred message and to take randomly one of them as the output. We note that the performance of the heuristic retrieval by boosting almost coincides with the brute-force ML decoding in both cases. The curves of the accelerated version (i.e. the boosting is only performed on the cluster that contains the largest number of activated nodes) approach them especially in the range of low error rates.

5.2 Messages with Large Number of Insertions

We also assess the performance of boosting methods in sparse networks when there is a large number of insertions. In Fig. 4, $M = 80000$ random messages of order $c = 10$ are stored in a network of $\chi = 100$ clusters of $l = 64$ nodes each. Note that the density of the resulting network is approximately 16.8%. We compare three algorithms: c-GWsTA, LsKO and LsKO by boosting by varying

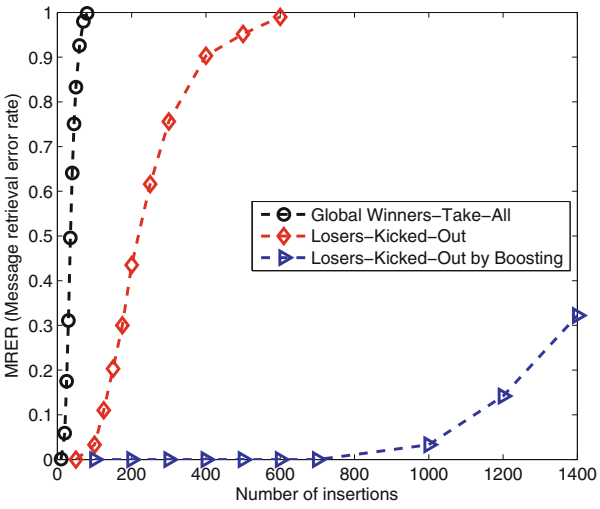


Fig. 4. Comparison of selection of winners algorithms in presence of a large number of insertions. The curves are illustrated for three algorithms: GWsTA, LsKO and LsKO by boosting. Simulation parameters are as follows: $\chi = 100$, $l = 64$, $M = 80000$, $c = 10$. The density of the resulting network is approximately 16.8%.

the number of insertions c_{ins} from 0 to 1400 (that is, 140 times the original message order). For the simulations of the algorithm LsKO by boosting, the optional pruning step is not applied and the retrieval order threshold c' is fixed to c. The algorithm c-GWsTA has relatively poor performance when the number of insertions increases: more than 80 insertions (8 times the message order) makes the retrieval of any message impossible, whereas LsKO performs considerably better in this scenario. And LsKO with boosting outperforms drastically both of them: the error rate is approximatively 3% when the network has to find the target clique of order 10 among a subgraph of 1010 nodes.

5.3 Complexity

Let us investigate the complexity of proposed algorithms. We suppose that the input message activates n_a nodes. The complexity of message passing is $\mathcal{O}(n_a^2)$ and that of selection of winners (LWTA, GWsTA or LsKO) is $\mathcal{O}(n_a)$. Thus, the complexity of a single iteration is $\mathcal{O}(n_a^2)$. The number of iterations before convergence is $\mathcal{O}(n_a)$. Finally, the complexity of classical retrieval algorithms is:

$$\mathcal{C} = \mathcal{O}(n_a^3).$$

For heuristic boosting solutions, the order of the subgraph in which we apply classical retrieval algorithms is $\mathcal{O}(dn_a)$. Thus, in a similar way as above, the

complexity of a single attempt of clique finding is $\mathcal{O}(d^3 n_a^3)$. The number of attempts is $\mathcal{O}(n_a)$. Finally, the complexity can be expressed as:

$$\mathcal{C}_{\text{boosting}} = \mathcal{O}(d^3 n_a^4).$$

Heuristic boosting solutions maintain thereby polynomial complexity. And if input message order n_a satisfies $n_a = \mathcal{O}(1/d^3)$, the complexity is comparable to that of classical algorithms.

6 Conclusion

In this paper, we proposed heuristic boosting algorithms to cope with challenging decoding scenarios where a heavily corrupted probe is present. These solutions not only outperform drastically previously existing algorithms in terms of retrieval error rate approaching the brute-force ML decoding performance, but also maintain polynomial complexity that assures efficient and biologically plausible implementation.

These algorithms open interesting perspectives to improve performance with real-world data in clique-based networks. For instance, authors of [16] have proposed to encode correlated messages by multiplying material resources for segments with frequent apparition. During retrieval process, this requires addressing in each cluster a set of nodes that represents the same alphabet instead of a single node in the case of uniformly distributed entries. Our proposed heuristic solutions could be useful in such scenarios. Another possible application would be to deal with approximate entries, such as in the case of typoglycemia in writing and reading processes: the reader can understand the meaning of words in a sentence even when the letters are scrambled.

References

1. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America **79**(8), 2554 (1982)
2. Hopfield, J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings of the National Academy of Sciences **81**(10), 3088–3092 (1984)
3. Willshaw, D.J.: Models of distributed associative memory, ph.D dissertation, Edinburgh University (1971)
4. Sommer, F.T., Palm, G.: Improved bidirectional retrieval of sparse patterns stored by Hebbian learning. Neural Networks **12**, 281–297 (1999)
5. Palm, G.: Neural associative memories and sparse coding. Neural Networks **37**, 165–171 (2013)
6. Gripon, V., Berrou, C.: Sparse neural networks with large learning diversity. IEEE Transactions on Neural Networks **22**(7), July 2011
7. Aliabadi, B.K., Berrou, C., Gripon, V., Jiang, X.: Storing sparse messages in networks of neural cliques. IEEE Transactions on Neural Networks and Learning Systems, May 2014

8. Jiang, X., Gripon, V., Berrou, C., Rabbat, M.: Storing sequences in binary tournament-based neural networks. arXiv: 1409.0334 IEEE Transactions on Neural Networks and Learning Systems (under revision)
9. Jiang, X., Gripon, V., Berrou, C.: Learning long sequences in binary neural networks. In: International Conference on Advanced Cognitive Technologies and Applications, Nice, France, pp. 165–170 (2012)
10. Aboudib, A., Gripon, V., Jiang, X.: A study of retrieval algorithms of sparse messages in networks of neural cliques. In: International Conference on Advanced Cognitive Technologies and Applications, Venice, Italy, pp. 142–146 (2014)
11. Ackley, D., Hinton, G.E.: A learning algorithm for Boltzmann machines. Cognitive Science **9**(1), 147–169 (1985)
12. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics **5**, 115–133 (1943)
13. Palm, G., Schwenker, F., Sommer, F.T., Strey, A.: Neural associative memories. Biological Cybernetics **36** (1993)
14. Gripon, V., Berrou, C.: Nearly-optimal associative memories based on distributed constant weight codes. In: Information Theory and Applications Workshop (ITA), February (2012)
15. Gripon, V., Jiang, X.: Mémoires associatives pour observations floues. In: Proceedings of XXIV-th Gretsi seminar, September (2013)
16. Boguslawski, B., Gripon, V., Seguin, F., Heitzmann, F.: Twin Neurons for Efficient Real-World Data Distribution in Networks of Neural Cliques. Applications in Power Management in Electronic circuits. IEEE Transactions on Neural Networks and Learning Systems, submitted to

On Structures with Emergent Computing Properties. A Connectionist versus Control Engineering Approach

Daniela Danciu(✉) and Vladimir Răsvan(✉)

Department of Automation and Electronics,
University of Craiova, 13, A.I. Cuza str., 200585 Craiova, Romania
{ddanciu,vrasvan}@automation.ucv.ro

Abstract. This paper starts by revisiting some founding, classical ideas for Neural Networks as Artificial Intelligence devices. The basic functionality of these devices is given by stability related properties such as the gradient-like and other collective qualitative behaviors. These properties can be linked to the structural – connectionist – approach. A version of this approach is offered by the hyperstability theory which is presented in brief (its essentials) in the paper. The hyperstability of an isolated Hopfield neuron and the interconnection of these neurons in hyperstable structures are discussed. It is shown that the so-called "triplet" of neurons has good stability properties with a non-symmetric weight matrix. This suggests new approaches in developing of *Artificial Intelligence* devices based on the triplet interconnection of elementary systems (neurons) in order to obtain new useful emergent collective computational properties.

Keywords: Hyperstability · Triplet connection · Gradient behavior · Non-unique equilibrium points · Emergent properties · Hopfield neuron

1 Basic Ideas and Overview

In a paper presented at a previous event of this conference series [1] we cited two interesting references from the point of view of the structure analysis of the *Artificial Intelligence* (AI) devices [2,3]. Considering the very general starting and motivating issues, two founding backgrounds of the AI structures were in fact pointed out within that survey: the emergence of new computational properties and the stability. The two theses that structured the survey are: i) to infer networked repetitive structures (similar, possibly to Artificial Neural Networks) that might be problem oriented or of general purpose and ii) to demonstrate specific global convergence to some steady states accounting for the aforementioned algorithmic architectures. These theses will underline the present paper, too. In order to strengthen this, we shall refer in the following to one of the classics of the field, namely J. J. Hopfield [4,5]. In order to be very explicit in presenting the idea that almost all basic background and paradigms of the

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 415–429, 2015.
DOI: 10.1007/978-3-319-19258-1_35

Neural Networks arise from these two references, we shall reproduce here some short but very relevant citations of [4,5].

"In physical systems made from a large number of simple elements, interactions among large numbers of elementary components yield collective phenomena such as the stable magnetic orientations and domains in a magnetic system or the vortex patterns in fluid flow. Do analogous collective phenomena in a system of simple interacting neurons have useful "computational" correlates? For example, are the stability of memories, the construction of categories of generalization, or time-sequential memory also emergent properties and collective in origin?" [4]

"Consider a physical system described by many coordinates $X_1, \ldots, X_N$*, the components of a state vector* $\bar{X}$*. Let the system have locally stable limit points* $\bar{X}_a, \bar{X}_b, \ldots$*. Then, if the system is started sufficiently near any* $\bar{X}_a$*, as at* $\bar{X} = \bar{X}_a + \bar{\Delta}$*, it will proceed in time until* $\bar{X} \approx \bar{X}_a$*. We can regard the information stored in the system as the vectors* $\bar{X}_a, \bar{X}_b, \ldots$*. The starting point* $\bar{X} = \bar{X}_a + \bar{\Delta}$ *represents a partial knowledge of the item* $\bar{X}_a$ *and the system then generates the total information* $\bar{X}_a$*. Any physical system, whose dynamics in phase space is dominated by a substantial number of locally stable states to which it is attracted, can therefore be regarded as a general content addressable memory. The physical system will be a potentially useful memory if, in addition, any prescribed set of states can readily be made the stable states of the system."* [4]

"All our interesting results arise as consequences of the strong back-coupling." [4]

This last citation deserves an "early" comment: in contrast to feedforward networks like Perceptrons, the considered interesting objects contain *feedback connections*. As mentioned in [6], feedback may be an instability generator and among the mechanisms that generate instability one should "discover" some hidden feedback. However, as it will appear in the following, the existence of unstable steady states is not incompatible with the existence of many stable states – it is in fact a coexistence of both states within the state space. But let Hopfield to continue:

"Our model, in contrast (to the linear structures - our remark), will use its strong nonlinearity to make choices, produce categories, and regenerate information and, with high probability, will generate the output from confusing mixed stimulus." [4]. And to continue along the same direction of the nonlinear structures:

"A study of emergent collective effects and spontaneous computation must necessarily focus on the nonlinearity of the input-output relationship. The essence of computation is nonlinear logical operations. The particle interactions that produce true collective effects in particle dynamics come from a nonlinear dependence of forces on positions of the particles." [4]. Further, new specific problems are stated:

"We will therefore initially assume that such a T_{ij} *("synapse intensity" parameter - our remark) has been produced by previous experience (or inheritance)."* [4]

Obviously here the standard framework of considering the learning dynamics separate from network own dynamics is to be met. Another specific technical problem, frequently met in further development reads as follows

"Why should stable limit points or regions persist when $T_{ij} \neq T_{ji}$?" [4]. This symmetry assumption appears as a direct consequence of the energy-like Lyapunov functions that are currently in use.

"The original model behaves as an associative memory (or CAM) when the state space flow generated by the algorithm is characterized by a set of stable fixed points. If these stable points describe a simple flow in which nearby points in state space tend to remain close during the flow (i.e., a non-mixing flow), then initial states that are close (in Hamming distance) to a particular stable state and far from all others will tend to terminate in that nearby stable state." [5] Here we can observe the idea of having uniqueness of the solutions for the system of ordinary differential equations – the "non-mixing" – as well as the convergence to some equilibria acting as attractors being thus asymptotically stable. The author of [4] continue

"Convergent flow to stable states is the essential feature of this CAM (Content Addressable Memory) operation." [5]

"Together with the boundedness of E, Eq. 10 (i.e. the inequalities for the derivative of the energy function E long the solutions) shows that the time evolution of the system is a motion in state space that seeks out minima in E and comes to a stop at such points. E is a Lyapunov function for the system." [5]

It thus appears that almost all paradigms acting in neural networks dynamics may be found *in nuce* (in a nutshell) in the seminal papers of Hopfield. Even the introduction of a high gain λ, that makes the sigmoid function steeper, and its application in the analysis and design of KWTA networks is to be met in [5]. We shall develop this idea in the next section.

2 Two Theses of the Present Paper

Summarizing the outcomes of the previous section we can point out the following ideas:

i) To ensure functionality of a quite large class of Artificial Intelligence devices it is necessary to obtain a set containing sufficiently numerous equilibria which should be asymptotically stable;
ii) This set of asymptotically stable equilibria is not an outcome of the design process i.e. this is not intentional from the point of view of the network.

During almost 15 years we followed in our papers these ideas of the dynamics of neural networks of various structures containing feedback interconnections. Our main aim is to find what we call "the basic functional validation" of an AI device, in particular of neural networks designed for various purposes: the *gradient-like behavior* of the associated dynamical system which, as it has been seen, is a nonlinear system with several equilibria. Taking into account that

ensuring this qualitative behavior (the gradient behavior) is not the goal of the functional design, it appears as an *a posteriori* task. We cite but a few of our papers illustrating this thesis [7–10] and aiming at obtaining conditions ensuring useful qualitative properties among which the gradient-like behavior is the best suited – see also [11,12]. Worth mentioning that our papers are not singular as research orientation. We may cite here [13–15] and [16–18], too. It is felt however that in our papers we pointed out more explicitly the aforementioned qualitative properties (which are summarized in the next section).

In order to ensure this qualitative analysis, we made use of the global qualitative theory of the dynamical systems with several equilibria [19–21]. The best analysis instrument for this study turned to be the second method of Lyapunov, at its turn based on the so-called *natural Lyapunov functions* [22,23]. Among them, we may cite the "neural energy function" introduced (again) by Hopfield himself and the Cohen-Grossberg function which is much alike to other natural Lyapunov functions occurring in chemical kinetics, biology and, possibly, in other competitive/cooperative structures of dynamical systems.

Summarizing this section, the following can be stated.

- The performance of the neural networks as well as of other Artificial Intelligence devices is conditioned by some "collective behavior" of many equilibria, among which the best is the *gradient-like behavior*. Usually, this condition is overlooked in the functional design phase and has to be checked *a posteriori*.
- The most efficient (and the most common) way of seeking the gradient-like behavior is to make use of some Lyapunov-like lemmas which are obtained from the Barbašin-Krasovskii-LaSalle invariance principle [19–21].

We can see that ensuring gradient-like behavior for the neural networks (as well as for other AI devices) from the early design stage is still a *desideratum*/wishful thinking. Nevertheless several attempts do exist and we send the reader to e.g. [24,25]. A rather different approach that has been never involved in the field of the Artificial Intelligence – with a notable exception [26] – is the *Theory of hyperstability* elaborated by V. M. Popov [27]. In his preface to all editions of this seminal book (Romanian, 1966; Russian, 1970; French and English, 1973) the author states that he *"starts from the usual point of view of the control engineer who likes to have at his disposal a wide range of elements capable of being combined in various ways to form control systems as complex as desired, but who does not like to burden his creative imagination with instability problem"*. It is felt that this approach is suitable for our purpose if we replace in the above sentence "control" by "Artificial Intelligence" and "stability" by e.g. "gradient-like behavior".

Consequently, what remains of this paper is structured as follows. We recall first the main concepts and tools from the theory of systems with non-unique equilibria – definitions and Lyapunov-like lemmas. Next, we discuss the basic concepts and structures of the theory of hyperstability with particular reference to a less known hyperstable structure proposed by Popov in a less circulated work [28] – the "triplet". We then consider the Hopfield analog neuron independently as well as connected in a triplet of neurons in order to point out

hyperstability properties. In this way we try to complete the list of main hyperstable blocks drawn by Popov himself ([27], Appendix D). Finally, we make an attempt to deduce emergent properties of the triplet from its hyperstability properties and end the paper with some conclusions and sketch a research programme.

3 Qualitative Properties of the Systems with Non-Unique Equilibrium State

We shall make use here of the basic definitions and results from [19–21], some of which are to be found also in [23]. Worth mentioning that the notions from the definition, while initially formulated for systems of ordinary differential equations, may be easily reformulated for other kinds of equations e.g. those with delayed argument.

Definition 1. *Consider the autonomous system of ordinary differential equations on* $\mathbb{R}^n$

$$\dot{x} = f(x) \ , \ x \in \mathbb{R}^n \tag{1}$$

with $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ *being a continuous locally Lipschitz mapping.*

1° *Any constant solution* c *of (1) is called* equilibrium. *The set of equilibria* $\mathcal{E}$ *is called* stationary set.

2° *A solution of (1) is called* convergent *if it approaches asymptotically some equilibrium:* $\lim_{t\to\infty} x(t) = \bar{x} \in \mathcal{E}$. *A solution is called* quasi-convergent *if it approaches asymptotically the stationary set:* $\lim_{t\to\infty} d(x(t), \mathcal{E}) = 0$.

3° *System (1) is called* monostable *if every bounded solution is convergent; it is called* quasi-monostable *if every bounded solution is quasi-convergent.*

4° *System (1) is called* gradient-like *if every solution is convergent; it is called* quasi-gradient-like *if every solution is quasi-convergent.*

It must be mentioned here that there exist several equivalent notions to the aforementioned ones; the reader can find an account on the subject in [20,23]. Establishing the properties from Definition 1 is possible by making use of the following result that, at its turn, may be obtained from the Barbašin-Krasovskii-LaSalle invariance principle [20,23].

Lemma 1. 1° *Consider the nonlinear system (1) and assume existence of a continuous function* $V : \mathbb{R}^n \to \mathbb{R}$ *which is non increasing along any solution of the system. If, additionally, a bounded on* $\mathbb{R}^+$ *solution* $x(t)$ *for which there exists some* $\tau > 0$ *such that* $V(x(\tau)) = V(x(0))$ *is an equilibrium, then the system is quasi-monostable.*

2° *If the assumptions of* 1° *hold and either* $V(x) \to \infty$ *for* $|x| \to \infty$ *or all solutions of the system are bounded, then the system is quasi-gradient-like.*

3° *If the assumptions of* 2° *hold and the set* $\mathcal{E}$ *is discrete (i.e. it consists of isolated equilibria only) then the system is gradient-like.*

We send the reader to [7] (Section "The Extension of the LaSalle Like Theory") for the version of the above lemma in the case of the time-delay systems with non-unique equilibrium points.

4 Elements of Hyperstability Theory

A. We shall restrict ourselves to the case of dynamics described by ordinary differential equations while the main reference [27] lists also other types of dynamics. The basic element of this theory is the structure "differential equation + integral system" – later called *Popov system* (system in the sense of Popov):

$$\dot{x} = f(x, u(t), t), \quad u \in \mathbb{C}^m, \quad x \in \mathbb{C}^n \tag{2}$$

$$\eta(t_0, t) = \phi(x(\tau), \tau)\,\big|_{t_0}^{t} + \int_{t_0}^{t} \psi(x(\tau), u(\tau), \tau)\mathrm{d}\tau \tag{3}$$

under the following basic assumptions: i) $f : \mathbb{C}^m \times \mathbb{C}^m \times \mathbb{R} \to \mathbb{C}^n$ is continuous in all arguments thus verifying an existence theorem for (2) e.g. the Peano existence theorem; ii) $\psi(\cdot, \cdot, \cdot)$ is such that along the solutions of (2) it is integrable (e.g. in the sense of Lebesgue).

Definition 2. *A set of functions composed of*

1. *a piecewise continuous function* $u : [t_0, T_0] \to \mathbb{C}^m$*;*
2. *an absolutely continuous function* $x : [t_0, T_0] \to \mathbb{C}^n$ *where* x_0 *is given;*
3. *an absolutely continuous function* $\eta(t_0, \cdot) : [t_0, T_0] \to \mathbb{R}$ *defined for* $t \in [t_0, T_0]$ *along the values of the aforementioned functions* $u(\cdot)$ *and* $x(\cdot)$

is called a solution *of the system (2)-(3).*

Definition 3. *System (2)-(3) is called hyperstable (in the strict sense) if there exist two Kamke-Massera functions* $\alpha : \mathbb{R}^+ \to \mathbb{R}^+$ *and* $\beta : \mathbb{R}^+ \to \mathbb{R}^+$ *(continuous,* $\alpha(\rho) \geq 0$*,* $\beta(\rho) \geq 0$*;* $\alpha(0) = \beta(0) = 0$*) with* $\alpha(\cdot)$ *being strictly increasing and global i.e.* $\lim_{\rho \to \infty} \alpha(\rho) = +\infty$ *such that*

$$\alpha(|x(t)|) \leq \eta(t_0, t) + \beta(|x(t_0)|), \quad \forall t \geq t_0 \tag{4}$$

along the solutions of (2)-(3).

We mention that this is not the most general definition introduced by Popov [27]: in fact, it arises as a special case of a sufficient condition of hyperstability (which nevertheless turns to be necessary and sufficient in most applications).

If we add to (2)-(3) a vector relation of the form

$$h(x(t), u(t), t) = 0 \tag{5}$$

where $h : \mathbb{C}^n \times \mathbb{C}^m \times \mathbb{R} \to \mathbb{C}^p$ is a continuous application in all arguments, the aforementioned definitions still hold provided we add to **Definition** 2 (the definition of the solution) the condition of verifying (5).

Another notion is the "sum" of two Popov systems defined as follows

$$\dot{x}_j = f_j(x_j, u_j(t), t) \tag{6}$$

$$\eta_j(t_0,t)=\phi_j(x_j(\tau),\tau)\,\big|_{t_0}^{t}+\int_{t_0}^{t}\Psi_j(x_j(\tau),u_j(\tau),\tau)\mathrm{d}\tau,\quad j=1,2 \tag{7}$$

$$\eta(t_0,t)=\eta_1(t_0,t)+\eta_2(t_0,t). \tag{8}$$

In fact the differential equation of the sum-system is given by (6) – as Cartesian product of equations – and by the index (8) which is the sum of the two indices. The following result holds [27]

Proposition 1. *The sum of two hyperstable systems is a hyperstable system.*

Obviously this result can be extended by induction to an arbitrary finite number of systems. In fact the result is even more general [27] but it is outside the interest of the present paper.

B. The next important notion is that of *block* (in the sense of Popov). If we associate to (2) a readout map – the output

$$v=g(x,u,t) \tag{9}$$

where $g:\mathbb{C}^n\times\mathbb{C}^m\times\mathbb{R}\to\mathbb{C}^m$ is a continuous mapping (in all arguments), the couple (2)-(9) will define a "square" block since the number of input terminals equals the one of the output terminals. Sometimes one may add to (2)-(9) some restrictions of the form

$$h(x,u,v,t)=0. \tag{10}$$

To each block of the form (2)-(9) one can associate a "system" of the form

$$\begin{gathered}\dot{x}=f(x,u(t),t)\\ \eta(t_0,t)=\Re e\left(\int_{t_0}^{t}u^*(\tau)v(\tau)\mathrm{d}\tau\right)=\Re e\left(\int_{t_0}^{t}u^*(\tau)g(x(\tau),u(\tau),\tau)\mathrm{d}\tau\right)\end{gathered} \tag{11}$$

where $2\Re e(u^*v)=u^*v+v^*u$.

In this way all notions defined for the Popov systems can be projected on the blocks i.e. solution, restrictions, hyperstability. Concerning the blocks, we have the following

Proposition 2. *Assume that the block (2)-(9) is hyperstable in the sense of Definition 3. Then among the free state transitions defined by the differential equation*

$$\dot{x}=f(x,0,t) \tag{12}$$

there is found the trivial equilibrium $x(t)\equiv 0$ which is stable in the sense of Lyapunov and all free state trajectories are globally bounded.

The proof is straightforward but we just mention that from this Proposition it follows in a necessary way that $f(0,0,t)\equiv 0$.

C. We shall consider now the block interconnections, starting with the following statement which is more than a conjecture.

Statement 1. *Those interconnections of hyperstable blocks to which one can associate sum systems generate hyperstable aggregated blocks.*

In fact, again the results of [27] are more general and more comprehensive but we shall restrict to the aforementioned case. In order to illustrate the aforementioned Statement, we shall follow [27] and discuss the basic interconnections (Fig. 1): feedforward and negative feedback.

For the feedforward connection (Fig. 1(a)) the interconnection relations are defined by

$$u_1 = u_2 = u, \;\; v = g_1(x_1, u, t) + g_2(x_2, u, t) \tag{13}$$

and the interconnection itself can be described as follows: the functions u_j, x_j, v_j, $(j = 1, 2)$, u, v are solutions of the "feedforward (parallel)" aggregated block $\mathbf{B_p}$ if they verify the equations

$$\dot{x}_j = f_j(x_j, u_j, t), \;\; v_j = g_j(x_j, u_j, t), \;\; j = 1, 2 \tag{14}$$

with the restrictions $u_1 = u_2 = u$, $v = v_1 + v_2$. We shall then have $2\Re e(u^* v) = \Re e(u_1^* v_1) + \Re e(u_2^* v_2)$ hence,

$$\eta(t_0, t) = \eta_1(t_0, t) + \eta_2(t_0, t) \tag{15}$$

Consider now the feedback interconnection (Fig. 1(b)). In the general – nonlinear – case the theory relies on the possibility of eliminating v_1 between the relations

$$v_1 = g_1(x_1, u - v_2, t), \;\; v_2 = g_2(x_2, v_2, t) \tag{16}$$

to obtain a relation of the form

$$v_2 = g_0(x_1, x_2, u, t). \tag{17}$$

The aggregated feedback block $\mathbf{B_f}$ is then defined by

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, u - g_0(x_1, x_2, u, t), t) \\ \dot{x}_2 &= f_2(x_2, g_1(x_1, u - g_0(x_1, x_2, u, t), t,), t) \\ v &= g_1(x_1, u - g_0(x_1, x_2, u, t), t). \end{aligned} \tag{18}$$

As previously, the interconnection can be described as follows: the functions u_j, x_j, v_j, $(j = 1, 2)$, u, v are solutions of the feedback aggregated block $\mathbf{B_f}$ if they verify (14) with the restrictions $u_1 + v_2 = u$, $u_2 = v - 1$, $v = v_1$. We shall then have

$$\Re e(u^* v) = \Re e((u_1 + v_2)^* v_1) = \Re e(u_1^* v_1) + \Re e(v_2^* u_2)$$

hence we have again (15). Worth mentioning that existence of (17) is a necessary condition for the aggregated dynamics to be well-defined. The fulfilment of this condition globally is quite a problem in the most general case. It holds locally provided the conditions of the implicit functions theorem are fulfilled.

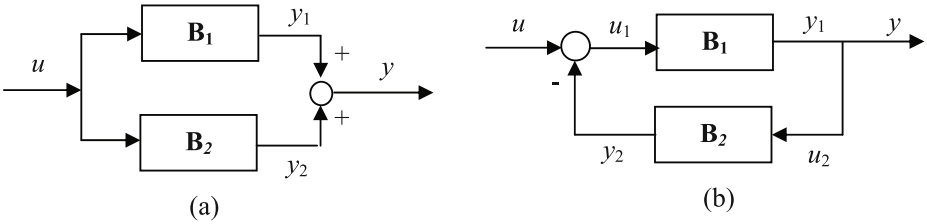


Fig. 1. The feedforward (parallel) interconnection (a) and the negative feedback interconnection (b) of two blocks

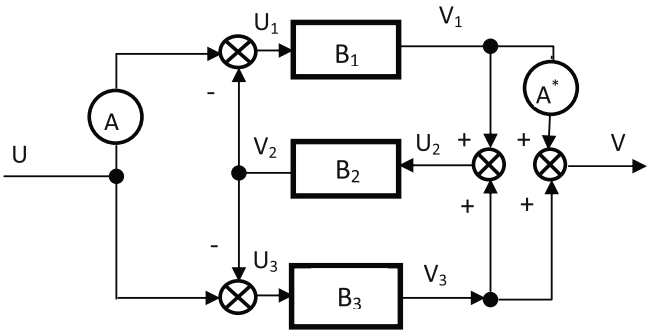


Fig. 2. The triplet interconnection

5 The Triplet Interconnection

In the analysis of the interconnections of hyperstable blocks ([27], page 135) one will find the following remark: "The parallel and negative feedback connections are not the only ones with the property that the resulting block is hyperstable whenever the component blocks are hyperstable."

In a less circulated paper [28], V.M. Popov introduced a new interconnection – of three blocks – which, at least in the case of "simple" SISO (Single-Input-Single-Output) blocks turns to be independent of the other two – the parallel and the feedback ones. In fact, this connection was introduced within the context of real positive transfer function synthesis – a problem arising from *Circuit Theory*. Since real positivity of the transfer function is a necessary and sufficient condition for the hyperstability of the linear time invariant blocks, the connection with the *Hyperstability Theory* is obvious.

Indeed, the triplet (Fig. 2) contains three blocks instead of two and this would send to a more general algebraic context of ternary algebra (e.g. [29]); a more application oriented point of view arising from the *Classical Control Theory* would consider the triplet in the context of the signal flow graphs. We shall however restrict ourselves to the hyperstability analysis, but deal with MIMO (Multi-Input-Multi-Output) blocks. As it is customary in the hyperstability

theory, square blocks with complex coefficients will be considered; also, the matrix A in Fig. 2 will be considered non-singular. The three component blocks are described in the most general case by

$$\dot{x}_j = f_j(x_j, u_j, t), \;\; v_j = g_j(x_j, u_j, t), \;\; j = 1, 2, 3. \tag{19}$$

The triplet interconnection is described by

$$u_1 = Au - v_2, \;\; u_2 = v_1 + v_3, \;\; u_3 = u - v_2, \;\; v = A^* v_1 + v_3. \tag{20}$$

We skip the problem of the well-posedness conditions (see the case of the feedback interconnection) and describe the aggregated block $\mathbf{B_t}$ in terms of the solutions: the functions $x_j, u_j, v_j, (j = 1, 2, 3), u, v$ are solutions of the "triplet aggregated" block $\mathbf{B_t}$ if they verify (19) with the restrictions (20). We shall then have

$$\begin{aligned}
\Re e(u^* v) &= \Re e(u^*(A^* v_1 + v_3)) = \Re e((u_1 + v_2)^*(A^*)^{-1}(A^* v_1 + v_3)) \\
&= \Re e(u_1^* v_1) + \Re e(v_2^* v_1) + \Re e((u_1 + v_2)^*(A^*)^{-1} v_3) \\
&= \Re e(u_1^* v_1) + \Re e(v_2^*(u_2 - v_3)) + \Re e((u_1 + v_2)^*(A^*)^{-1} v_3) \\
&= \Re e(u_1^* v_1) + \Re e(u_2^* v_2) + \Re e((u_1 + v_2)^*(A^*)^{-1} - v_2^*) v_3) \\
&= \Re e(u_1^* v_1) + \Re e(u_2^* v_2) + \Re e((u^* A^*(A^*)^{-1} - v_2) v_3) \\
&= \Re e(u_1^* v_1) + \Re e(u_2^* v_2) + \Re e(u_3^* v_3)
\end{aligned}$$

hence

$$\eta(t_0, t) = \eta_1(t_0, t) + \eta_2(t_0, t) + \eta_3(t_0, t) \tag{21}$$

6 The Triplet of Hopfield Neurons

The equations of an isolated Hopfield neuron can be found in the original paper [5]

$$\dot{x}(t) = -ax + wy + I \;, \;\; y = f(x) \tag{22}$$

with $a < 0, w > 0, f : \mathbb{R} \to \mathbb{R}, f(0) = 0$ being a sigmoid function i.e. strictly increasing, Lipschitz continuous and, sometimes, odd and/or globally defined. It is not difficult to see that (22) defines a feedback system of the type that may be found in the theory of hyperstability: a linear dynamic block and a nonlinear static one. This interconnection defines a hyperstable feedback structure. This can be shown following the approach of [27] (Chapter 5, § 25.8). Equations (22) are replaced by the following equivalent system of two dynamical blocks connected in feedback:

$$\begin{aligned}
\dot{x} &= -ax - w\mu_1 \\
\nu_1 &= (\alpha - a\beta)x + (\alpha/\bar{\Phi} - \beta w)\mu_1 \\
\beta\dot{\xi} + \alpha\xi - (\alpha/\bar{\Phi})\Phi(\xi) &= \mu_2 \\
\nu_2 &= \Phi(\xi) \\
\mu_1 = -\nu_2, \;\; \mu_2 &= \nu_1
\end{aligned} \tag{23}$$

where we assumed the sigmoid function confined to the sector $0 \leq \Phi(\xi)/\xi \leq \bar{\Phi}$. Here $\alpha > 0$ and $\beta \in \mathbb{R}$ are arbitrary parameters.

The nonlinear block of (23) is hyperstable. Indeed we have

$$\begin{aligned} &-\frac{1}{2}\beta\bar{\Phi}\xi^2(\tau)\,|_{t_0}^{t} + \int_{t_0}^{t} \mu_2(\tau)\nu_2(\tau)\mathrm{d}\tau \\ &= \beta\Psi(\xi(\tau))\,|_{t_0}^{t} + \alpha\int_{t_0}^{t} \phi(\xi(\tau))\left(\xi(\tau) - \frac{1}{\bar{\Phi}}\Phi(\xi(\tau))\right)\mathrm{d}\tau \end{aligned} \tag{24}$$

where

$$\Psi(\xi(\tau)) = \int_0^{\xi(\tau)} \left(\Phi(\lambda) - \bar{\Phi}\lambda\right)\mathrm{d}\lambda \leq 0.$$

The last inequality follows from the sector condition as well as the inequality

$$\Phi(\xi)(\xi - (1/\bar{\Phi})\Phi(\xi)) \geq 0.$$

We then deduce from (24) the inequality

$$\beta\Psi(\xi(t_0)) + \eta(t_0, t) \geq \beta\Psi(\xi(t)) \tag{25}$$

where we defined

$$\eta(t_0, t) \equiv -\frac{1}{2}\beta\bar{\Phi}\xi^2(\tau)\,|_{t_0}^{t} + \int_{t_0}^{t} \mu_2(\tau)\nu_2(\tau)\mathrm{d}\tau$$

If $\beta < 0$, the inequality (25) is nothing more but the hyperstability of the nonlinear block of (23). The hyperstability of the feedback interconnection follows from the hyperstability of the linear block whose transfer function reads

$$H(s) = \frac{1}{\bar{\Phi}} - \frac{(\alpha + \beta s)w}{s + a}, \quad \beta < 0, \quad w > 0, \quad a > 0. \tag{26}$$

Since there is a single pole $-a < 0$, this function is real positive provided

$$\frac{1}{\bar{\Phi}} - w \cdot \Re e\frac{\alpha + \imath\omega\beta}{a + \imath\omega} > 0. \tag{27}$$

If $\beta < 0$, this inequality holds provided $(\bar{\Phi}w)/a < 1$ i.e. the well known small gain condition.

If $\beta > 0$, we obtain from the same structure (23) the hyperstability of the nonlinear block

$$\beta\Psi(\xi(t_0)) + \eta(t_0, t) \geq \beta\Psi(\xi(t)) \tag{28}$$

where, this time we have

$$\Psi(\xi(\tau)) = \int_0^{\xi(\tau)} \Phi(\lambda)\mathrm{d}\lambda \geq 0$$

again from the sector condition while now

$$\eta(t_0, t) = \int_{t_0}^{t} \mu_2(\tau)\nu_2(\tau)\mathrm{d}\tau. \tag{29}$$

From (27) we obtain the best possible inequality $(\bar{\Phi}w)/a < 1$ (the small gain condition) provided we choose $0 < \beta/\alpha \leq 1/a$.

It may happen to obtain an improvement of the small gain condition by using the monotonicity condition for the sigmoid nonlinearity or by introducing local negative feedback for the isolated neuron i.e. $w < 0$ but these aspects are outside the mainstream of this paper.

B. We shall now write down the triplet interconnection of three Hopfield neurons. Starting from the equations of the three neurons

$$\begin{gathered} \dot{x}_i = -a_i x_i + w_{ii} f_i(x_i) + I_i, \quad i = 1, 2, 3 \\ \nu_i = f_i(x_i) \end{gathered} \tag{30}$$

and taking into account the structure of Fig. 2, we obtain the following equations

$$\begin{aligned} \dot{x}_1 &= -a_1 x_1 + w_{11} f_1(x_1) - w_{12} f_2(x_2) + I_1 + \alpha u(t), \\ \dot{x}_2 &= -a_2 x_2 + w_{21} f_1(x_1) + w_{22} f_2(x_2) + w_{23} f(x_3) + I_2, \\ \dot{x}_3 &= -a_3 x_3 - w_{32} f_2(x_2) + w_{33} f_3(x_3) + I_3 + \alpha u(t), \\ \nu &= \alpha f_1(x_1) + f_3(x_3). \end{aligned} \tag{31}$$

In this structure the biases I_i are used to ensure various input signals to be processed but also a sufficiently large set of equilibria. The overall terminals $u(t)$ and $v(t)$ allow interconnections to other structures. Remark that the weight matrix reads

$$W = \begin{pmatrix} w_{11} & -w_{12} & 0 \\ w_{21} & w_{22} & w_{23} \\ 0 & -w_{32} & w_{33} \end{pmatrix} \tag{32}$$

and obviously is not symmetric. Nevertheless, stability of this structure is ensured. If the frequency domain inequalities are satisfied strictly (with strict inequality, i.e. > 0) then this stability is also asymptotic [27], Chapter 5.

Even this extension to an asymmetric weight matrix is enough to show that the use of hyperstable structures, in particular the triplet, in Artificial Intelligence may prove to be extremely fruitful.

7 Some Conclusions. A Research Program.

This paper has its starting points in the classical papers of Hopfield [4,5] on emergent capabilities of the neural networks; it became clear since then that these properties rely on the existence of many equilibria that possess some stability-like properties among which gradient behavior is the best suited. At the same time, we observed that it would be useful, from the practical point of view, to be able to ensure such properties from the very beginning, in the design phase. The theory of hyperstability, conceived and developed by V. M. Popov in the 60ies and 70ies of the past century, is able to offer such a "connectionist"

point of view since it operates with such structures for which the hyperstability property of the components is conserved for the structure. The less known and exploited of these structures – the "triplet" – turned to be generating a new basic Neural Network cell, with good stability properties and with non-symmetric weight matrix. Embedding this structure in a repetitive network might open new perspectives in building Artificial Intelligence devices; it is similar to the building of Cellular Neural Networks but starting from a different basic structure.

From the point of view of the involvement of the hyperstability theory, the things may be considered at their starting point. Indeed hyperstability was conceived mainly for stability analysis of a single equilibrium point (of an autonomous – free – system). It is thus necessary to make the next step: from the autonomous system to the emergent properties. This would require analysis and counting of the equilibria under constant inputs; this had been a rather peculiar problem in classical analysis some decades ago but for sure it knew further development.

Another step is to obtain the gradient-like behavior or at least the quasi-gradient one. The standard approach has already been indicated – the Lyapunov function approach. But how to associate a Lyapunov function to the triplet? The hyperstability theory had been conceived as an independent theory with respect to the Lyapunov theory. It contains however some procedures of associating Lyapunov functions to hyperstable structures and they are more useful than the simple application of the inverse theorems of the Lyapunov theory.

The achievement of this step-by-step program would signify only the establishing of neurons triplet as an AI capability. This can lead to new approaches in developing of Artificial Intelligence devices based on the triplet interconnection of elementary systems (neurons) in order to obtain new useful emergent collective computational properties.

Acknowledgments. This work was partially supported by the grant number 10C/ 2014, awarded in the internal grant competition of the University of Craiova.

References

1. Răsvan, V.: Reflections on neural networks as repetitive structures with several equilibria and stable behavior. In: Rojas, I., Joya, G., Cabestany, J. (eds.) IWANN 2013, Part II. LNCS, vol. 7903, pp. 375–385. Springer, Heidelberg (2013)
2. Graupe, D.: Principles of Artificial Neural Networks, Advanced Series on Circuits and Systems, vol. 6, 2nd edn. World Scientific, Singapore (2007)
3. Grossberg, S.: Self-organizing neural networks for stable control of autonomous behavior in a changing world. In: Taylor, J.G. (ed.) Mathematical Approaches to Neural Networks, pp. 139–197. Elsevier (1993)
4. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **79**(1), 2554–2558 (1982)
5. Hopfield, J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. USA **81**(5), 3088–3092 (1984)

6. Neymark, Y.I.: Dynamical systems and controlled processes (in Russian). Nauka, Moscow (1978)
7. Danciu, D., Răsvan, V.: Dynamics of neural networks - some qualitative properties. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 8–15. Springer, Heidelberg (2007)
8. Danciu, D., Răsvan, V.: Gradient like behavior and high gain design of KWTA neural networks. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 24–32. Springer, Heidelberg (2009)
9. Danciu, D.: Dynamics of neural networks as nonlinear systems with several equilibria. In: Porto, A.B., Pazos, A., Buno, W. (eds.) Advancing Artificial Intelligence Through Biological Process Applications, pp. 331–357. IGI Global Publisher (2009)
10. Danciu, D., Răsvan, V.: Systems with slope restricted nonlinearities and neural networks dynamics. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part II. LNCS, vol. 6692, pp. 565–572. Springer, Heidelberg (2011)
11. Danciu, D.: Systems with several equilibria. Applications to neural networks (in Romanian). Universitaria, Craiova Romania (2006)
12. Danciu, D.: Neural networks. Stability, synchronization, delays (in Romanian). Universitaria, Craiova Romania (2010)
13. Forti, M.: On global asymptotic stability of a class of nonlinear systems arising in neural network theory. J. Differ. Equ. **113**(1), 246–264 (1994)
14. Forti, M., Tesi, A.: A new method to analyze complete stability of pwl Cellular Neural Networks. Int. J. Bifurcation Chaos **11**(3), 655–676 (2001)
15. Marco, M.D., Forti, M., Grazzini, M., Pancioni, L.: Limit set dichotomy and convergence of cooperative piecewise linear neural networks. IEEE Trans. on Circuits and Systems I **58**(5), 1052–1062, May 2011
16. Atencia, M., Joya, G., Sandoval, F.: Spurious minima and basins of attraction in higher-order Hopfield networks. In: Mira, J., Álvarez, J.R. (eds.) IWANN 2003. LNCS, vol. 2686, pp. 350–357. Springer, Heidelberg (2003)
17. Qin, S., Xue, X.: Dynamical analysis of neural networks of subgradient systems. IEEE Trans. on Aut. Contr. **55**(10), 2347–2352 (2010)
18. Lu, W., Wang, J.: Convergence analysis of a class of nonsmooth gradient systems. IEEE Trans. on Circuits and Systems I **55**(11), 3514–3527, December 2008
19. Gelig, A.K., Leonov, G.A., Yakubovich, V.A.: Stability of systems with non-unique equilibria (in Russian). Nauka, Moscow (1978)
20. Leonov, G.A., Reitmann, V., Smirnova, V.B.: Non-Local Methods for Pendulum-Like Feedback Systems, Teubner-Texte zur Mathematik, vol. 132. B.G. Teubner Verlagsgeselllschaft, Stuttgart (1992)
21. Yakubovich, V.A., Leonov, G.A., Gelig, A.K.: Stability of Stationary Sets in Control Systems with Discontinuous Nonlinearities, Stability, Vibration and Control of Systems, Series A, vol. 14. World Scientific, Singapore (2004)
22. Halanay, A., Răsvan, V.: Applications of Liapunov Methods in Stability, Mathematics and its Applications, vol. 245. Kluwer Academic Publishers, Dordrecht (1993)
23. Răsvan, V.: Dynamical systems with several equilibria and natural Liapunov functions. Archivum Mathematicum **34**(1), 207–215 (1998)
24. Brockett, R.W.: Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. In: Proc. IEEE Conference on Decision and Control, pp. 799–803. IEEE Press (1988)
25. Cohen, M.: The construction of arbitrary stable dynamics in nonlinear neural networks. Neural Networks **5**, 83–103 (1992)

26. Warwick, K., Zhu, Q., Ma, Z.: A hyperstable neural network for the modelling and control of nonlinear systems. Sādhanā **25**(2), 169–180 (2000)
27. Popov, V.M.: Hyperstability of Control Systems, Die Grundlehren der mathematischen Wissenschaften, vol. 204. Springer Verlag, Berlin (1973)
28. Popov, V.M.: An analogue of electrical network synthesis in hyperstability (in Romanian). In: Proc. Symposium on Analysis and Synthesis of Electrical Networks. p. 9.1. No. III, Power Institute of Romanian Academy (1967)
29. Kurosh, A.G.: Lectures in General Algebra. Pure and Applied Mathematics Monographs. Pergamon Press, London (1965)

Deep Neural Networks for Wind Energy Prediction

David Díaz(✉), Alberto Torres, and José Ramón Dorronsoro

Departamento de Ingeniería Informática e Instituto de Ingeniería del Conocimiento, Universidad Autónoma de Madrid, Madrid, Spain
david.diazv@estudiante.uam.es, {alberto.torres,jose.dorronsoro}@uam.es

Abstract. In this work we will apply some of the Deep Learning models that are currently obtaining state of the art results in several machine learning problems to the prediction of wind energy production. In particular, we will consider both deep, fully connected multilayer perceptrons with appropriate weight initialization, and also convolutional neural networks that can take advantage of the spatial and feature structure of the numerical weather prediction patterns. We will also explore the effects of regularization techniques such as dropout or weight decay and consider how to select the final predictive deep models after analyzing their training evolution.

1 Introduction

Having had a big first impact around 1990, Multilayer Perceptrons (MLPs) started a mild decline after the second half of that decade. A particularly puzzling issue was the difficulty to build efficient MLPs with three or more layers, in spite of the fact that the backpropagation computation of the gradient of the MLP error function could be carried out in a rather straightforward fashion. The reason behind this was the vanishing gradient phenomenon [8] which in turn was in part a consequence of the inadequacy of weight initialization.

However, this changed radically with the seminal paper by G. Hinton and R. Salakhutdinov [12] that showed how an unsupervised, stacking scheme based on Boltzmann machines could yield a good initialization (or pretraining) of the weights of a many-layered MLP, that could be then efficiently fine-tuned by backpropagation. Shortly afterwards, Y. Bengio and his coworkers proposed a similar and somewhat simpler pretraining using stacked autoencoders [4]. This opened the way to the enormous attention that deep MLPs (i.e., MLPs with three or more layers) or, in general, deep learning, have received in the past years.

This attention has in turn resulted in a great simplification of the initial schemes of Hinton and Bengio and has brought many new procedures and ideas to the MLP field, such as new initializations, or the replacement of some of the initial MLP recipes, such as sigmoid activations or weight decay regularization, by new proposals like rectified linear unit (ReLU) activations [9] or dropout

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 430–443, 2015.
DOI: 10.1007/978-3-319-19258-1_36

regularization [18]. Moreover, the very large datasets and deep MLP parameters often rules out batch learning. This has resulted in a large emphasis on online learning, usually over minibatches of randomly selected patterns, with much work being devoted to the choice of learning rates (or how to avoid them) or momentum methods such as Nesterov's acceleration. This raises the issue of when to stop training, something rather straightforward in the batch training of classical MLPs if an adequate regularization and an efficient optimizer were used. Furthermore, once the previous ingredients are in place, the need to specialized (and costly) pretraining is less accute and several initialization methods have been proposed that result in the training of effective deep models. A good example of such a global approach is [19]. Another key ingredient in the successful applications of deep learning is the use of convolutional layers, that concatenate a purely convolutional sublayer that processes inputs using localized window filters, and a pooling sublayer that aggregates the outputs of the previous sublayer. Starting with the work of Y. LeCun in the late 1990's, this processing is particularly natural when inputs have a spatial structure, as it is the case with images, and it has led convolutional deep nets to achieve state-of-the-art results in problems such as MNIST [7] or ImageNet [14].

All these advances have made possible the effective training of very large deep networks with hundreds of thousands of weights which, in turn, makes imperative the use of software that can take advantage of high performance hardware endowed with parallelization (i.e., multicore machines) and vectorization (i.e., GPU units). Besides, the fast pace of change in the field and the fact that there is still not an accepted multipurpose architecture makes it quite difficult to work with self developed code; instead, it is well advised to rely on publicly available libraries and environments such as the Caffe [13] deep learning framework or the Pylearn2–Theano libraries [5] [3] [10] that we use here.

In any case, it seems that the bulk of deep learning research concentrates on computer vision, speech recognition and natural language processing problems. This is partially natural in view of the broad similarity between deep learning architectures and the processing hierarchies in the visual cortex [15] (although deep learning algorithms are quite different from the Spike-Timing-Dependent Plasticity learning rule most accepted in neurobiology). As such, deep learning algorithms are increasingly seen as representation learning procedures that yield at each layer increasingly more abstract representations in such a way that features in the higher layers capture possibly more powerful data features.

However, the successful exploitation of such a processing may also take place in simpler regression problems that, nevertheless, have input patterns with a spatial structure. The goal of this work is the prediction of wind energy production. Spain is among the world leaders in wind energy with a very high penetration that in some special days and hours can meet a very high percentage of Spain's electricity demand. Obviously, this high penetration makes it very important to provide accurate prediction of wind energy, with standard MLPs (usually at the farm level) and Support Vector Regression (SVR) (for large scale prediction) being the models of choice. The inputs for such models are the forecasts provided

by numerical weather prediction (NWP) systems such as the ECMWF [1] or the GFS [2]. These predictions are forecasts of several weather variables given at the points of a rectangular grid that covers the areas under study and that reflects some underlying orographic model. One may thus view an area wide NWP forecasts as a set of feature maps (the individual weather variables) having a spatial structure (that of the underlying geography) in much the same way that the RGB channels of an image correspond to feature maps with a two dimensional structure.

Under the previous scheme, the consideration of convolutional networks to derive wind energy forecast arises as a natural option and they will be one of the models considered in this paper. Our main purpose is to develop a methodology to build models that can provide accurate predictions from the original data with as little pre-processing and expert knowledge as possible. Given the very wide range of proposals in the literature, this implies we must make beforehand concrete choices of network initialization, online training procedure, activation function and regularization scheme. Of course, on top of this, a more or less general network architecture also has to be selected.

We will develop the choices we make in the next sections. Besides standard "small" MLPs and SVR models that we use as reference benchmarks, we will consider deep MLPs with a standard multilayer structure, general deep convolutional networks (CNNs) and also an adaptation of the well-known LeNet [16], one of the most successful architectures for character recognition. In all those deep nets we will use Glorot–Bengio weight initialization [8], ReLUs as activation functions [9], dropout regularization [18] complemented with standard weight decay in the final fully-connected layers, random mini-batch gradient descent over batches of moderate size and conjugate gradient as the training algorithm. This enables us to work with a fixed, fairly general learning rate, that is no longer a parameter to explore. Summing things up, our main contributions are:

- We review some of the latest proposals in DNNs and propose general guidelines to apply deep MLPs in regression problems.
- We thoroughly explore the application of the two main paradigms in DNNs to the problem of local and large scale wind energy prediction.
- We introduce a variant of the well known LeNet convolutional neural network adapted to wind energy prediction and show it to be very competitive with other DNN architectures or state of the art methods such as Support Vector Regression.

As mentioned before, we will use Pylearn2 [10]–Theano [3] [5] platform as it includes a wide variety of already tested neural networks and allows us to explore several of the latest and most effective proposals for deep network training. An important advantage of having Theano as the underlying numerical library is that we can exploit its capabilities for code execution on GPUs, something crucial given the network sizes and input dimensions we work with. We run our experiments on a machine equipped with a NVidia Tesla K40 GPU which makes possible reasonable execution times and, hence, the capability of exploring a fairly large number of deep model configurations.

The rest of the paper is organized as follows. In Section 2 we review our choices for deep network configuration and discuss some of its details. Section 3 contains a succinct description of the framework for wind energy prediction over NWP inputs, a description of our experimental setup and the prediction results for both the Sotavento wind farm and the entire wind energy prediction over peninsular Spain that is overseen by Red Eléctrica de España (REE). Finally, in Section 4 we briefly discuss our results and offer pointers to further work.

2 Deep Neural Networks

We briefly review here some of the key issues when configuring and training Deep Neural Networks.

2.1 Initialization

There have been several heuristic proposals for weight initialization in "classical" MLPs. For instance, a common choice is to take them from a uniform distribution $U\left[-\frac{1}{\sqrt{M}}, \frac{1}{\sqrt{M}}\right]$, with M the fan-in of the neuron, i.e., the number of weights feeding into it. However, it was found experimentally in [8] that in a deep MLP initialized in such a way, back-propagated gradients were progressively smaller when moving from the output layer towards the input layer and, in addition, their variances also decrease. In other words, backpropagating such an initialization may result in vanishing gradients in the first layers following the input and, thus, in a network which is insensitive to its inputs and unable to "learn" them.

The more detailed analysis in [17], also oriented to "classical" MLPs and where properly normalized hyperbolic tangents were used, pursued a goal of keeping the (linear) activations and (non linear) outputs of a neuron in the $[-1,\ 1]$ active range of the (normalized) hyperbolic tangent. Assuming inputs normalized to zero mean and unit variance component-wise, it is suggested in [17] to use an uniform distribution $U\left[-\frac{\sqrt{3}}{\sqrt{M}}, \frac{\sqrt{3}}{\sqrt{M}}\right]$. This analysis was extended in [8], where, assuming again the neuron outputs z_i of the i-th layer also to be in the $[-1,\ 1]$ active range, initialization should ensure first that $Var(z_i) \simeq Var(z_i')$ across the successive layers and also that $Var\left(\frac{\partial J}{\partial z_i}\right) \simeq Var\left(\frac{\partial J}{\partial z_i}\right)$, where J denotes the MLP cost function. This translates into the following equations for the initial weights W_i

$$M_i Var(W_i) = 1;\ M_{i+1} Var(W_i) = 1$$

where M_i and M_{i+1} are the fan-in and fan-out of the units in the i-th layer. An approximation to both is to take $Var(W_i) = \frac{2}{M_i + M_{i+1}}$, i.e., to initialize the W_i using an uniform distribution $U\left[-\frac{\sqrt{6}}{\sqrt{M_i+M_{i+1}}}, \frac{\sqrt{6}}{\sqrt{M_i+M_{i+1}}}\right]$. Note that when $M_i = M_{i+1}$, we get back the initialization proposed in [17].

We will use the initialization in [8] but working with Rectified Linear Unit (ReLU) activations, discussed next, instead of the hyperbolic tangent ones. While the rationale in [8] may not apply, the recent analysis in [11] of weight initialization for ReLU activation suggests to dilate the Glorot–Bengio uniform intervals by a factor of 1.5, and, in fact, we have observed that this usually yields better results.

2.2 Activation Function

As mentioned, we have used ReLUs for all hidden layer activations and linear units in the output layer. The ReLU transfer function is $r(x) = \max(0, x)$, that is, their response to the opposite of a positive excitatory input is just 0; in particular, ReLUs do not have a sign antisymmetry, as is the case with the hyperbolic tangent. On the other hand, ReLUs share some similarities with the functions relating neuronal input currents and firing rates that appear in the leaky integrate and fire models used in biological neuron models [9]. Besides, ReLUs induce sparsity in the representations of the successive layers; for instance, right after the uniform weight initialization, the outputs of about half the network neurons should be zero, as they would correspond to negative (inhibitory) inputs. This may partially explain the fact that ReLUs seem to be less affected than other activations by poor initializations. In any case, this point deserves further study.

2.3 Regularization

It is obvious that the extremely large number of weights in a deep MLP makes regularization mandatory to avoid overfitting. The standard regularization technique in classical MLPs is weight decay applied across all the layers; i.e., the square norm weight penalty considered for all layer weights is added to the MLP cost function. It has an obvious place in a last layer with linear outputs, as it performs ridge regression on the features induced in that last layer by the deep processing of the inputs.

However, for the other layers we will use dropout [18], that we briefly describe next. If a_i^l denotes the i-th activation of the l-layer and z_i^l the corresponding output, the standard feedforward processing would yield $z_i^l = f(a_i^l) = f(w_i^l z^{l-1} + b_i^l)$, where f is the activation function. However, with dropout, a 0–1 vector r^l is first generated applying a Bernoulli distribution componentwise. The feedforward process becomes

$$z_i^l = f(a_i^l) = f\left(w_i^l (z^{l-1} \odot r^l) + b_i^l\right),$$

where $\odot$ denotes the componentwise product. Each element in r^l has a probability p of being 1, so dropout can be seen as sub-sampling a larger network at each layer. The output errors are backpropagated as in standard MLPs for gradient computations and the final optimal weights w^* are downscaled as $w_f^* = pw^*$ to yield the final weights used for testing.

Dropout clearly induces a regularization of the network's weights. Moreover, it is reminiscent to the well known bagging technique for ensembles that repeatedly subsamples data to build specific models and then takes the average. However, in dropout all the "models" (i.e., the particular feedforward Bernoulli realizations) share weights and they are "trained" in a single step. Although we will not use it, in [18] it is also suggested that network performance improves when dropout is combined with a bound on the L_2 norm of the weights, i.e., when they are constrained as $\|w\|_2 \leq c$, with c a second tunable parameter on top of the Bernoulli probability p.

2.4 Convolutional Layers

Standard deep MLP architectures tend to favor layers with a high number of hidden units. This also leads to a high number of weights, $M \times M'$ if we fully connect an M unit layer with an M' one, a number that can become rather large if, for instance, inputs have a two dimensional structure, as it is the case with images. Convolutional layers arise in part as a way to avoid this by limiting the fan–in of a hidden unit to come from a localized subset of units in the previous layer. Of course, how to define such a restricted fan–in is, in general, problem–dependent, but when data have an intrinsic spatial structure a natural approach to localize the connections is to work over small patches.

More precisely, assume inputs or layer outputs to be one channel structured as an $M_1 \times M_2$ matrix, and consider in them $K \times K$ submatrix patches. They could be either disjoint or partially overlapping; we can parameterize this considering a stride value S that gives the displacement applied when we move horizontally and vertically from one patch to the next one. Assuming for simplicity a $S = 1$ stride, there are such $(M_1 - K + 1) \times (M_2 - K + 1)$ (overlapping) patches x_j. A first transform is to derive a patch feature $p_j = f(w * x_j + b)$ where f is the activation function, $*$ denotes the convolution operator between the $K \times K$ filter w and the patch x_j and b is the bias of the filter. This transforms an $M_1 \times M_2$ input X into an $(M_1 - K + 1) \times (M_2 - K + 1)$ convolutional output X' and usually a number L of filter pairs (w_l, b_l) (or of feature maps) have to be learned. Thus, the number of weights in a convolutional sub–layer is a rather modest $L \times K^2$ but, on the other hand, the output dimension would be $L \times (M_1 - K + 1) \times (M_2 - K + 1)$, which for $L > K$ might greatly increase the number of hidden units in the next layer. To curb this (and avoid a possible overfitting), a second pooling (or subsampling) sub–layer is applied in which an operation such as averaging or computing the max is applied on $P \times P$ patches of X' to derive the final output X_C of the convolution–pooling combined process; X_C has a $L \times (M_1 - K - P + 2) \times (M_2 - K - P + 2)$ dimension.

This combined convolution–pooling process is called a convolutional layer; it allows for a localized processing of the layer's input using a moderate number of weights (note that there are no weights in the pooling sub–layer) while arriving at a number of units in the next layer similar to that of the previous one. Of course, we stress again that, to be effective, a convolutional layer must act on inputs that have a spatial structure (such as images) and are naturally distributed in

feature channels (such as the RGB decomposition). This is also happens in our case, where weather prediction has an obvious spatial structure in which different meteorological features (pressure, temperature, wind components, etc.) can be seen as corresponding to different channels.

3 Experiments

In this section we will apply DNNs to the problem of predicting wind energy production, first on the Sotavento wind farm and then over peninsular Spain.

3.1 NWP and Production Data

We will work with the following eight meteorological variable forecasts given by the European Centre for Medium-Range Weather Forecasts (ECMWF) system for Numerical Weather Prediction (NWP):

- P, the pressure at surface level.
- T, the temperature at 2m.
- V_x, the x wind component at surface level.
- V_y, the y wind component at surface level.
- V, the wind norm at surface level.
- V_x^{100}, the x wind component at 100m.
- V_y^{100}, the y wind component at 100m.
- V^{100}, the wind norm at 100m.

In the Sotavento case they are taken on 15×9 rectangular grid centered on the Sotavento site (43.34°N, 7.86°W); input dimension in this case is thus $15 \times 9 \times 8 = 1,080$. For peninsular Spain we consider a 57×35 rectangular grid that covers entirely the Iberian peninsula; input dimension is now a very large $57 \times 35 \times 8 = 15,960$.

Wind energy data for Sotavento are publicly available; those for peninsular Spain were kindly provided by Red El?ctrica de Espa?a (REE). In both cases we normalize them to the $[0, 1]$ interval by dividing actual wind energy production by the maximum possible value in each case. We will work with data for the years 2011, 2012 and 2013, that we will use as training, validation and test subsets respectively. Since NWP forecasts are given every three hours, each subset will approximately have $(24/3) * 365 = 2,920$ patterns.

3.2 Deep Models

We will consider deep networks with either all their layers being fully connected, which we call deep MLPs, or with a number of initial convolutional layers followed by fully connected ones; we call these models deep convolutional neural networks, or deep CNNs. As reference models we will work with "standard" one hidden layer MLPs and also with Support Vector Regression (SVR) models,

Algorithm 1 Hyper-parameter search

procedure HYPER-PARAMETER SEARCH(n, m) ▷ $n \times m$ iterations
  randomly initialize an hyper–parameter vector p
  $p^* = p$ ▷ p^*: optimal hyper–parameter vector
  for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, m$ **do**
      $k \leftarrow$ random value in $\{1, \ldots, m\}$
      $p_k \leftarrow$ random value in $\{v_1^k, \ldots, v_{N_k}^k\}$
      evaluate the p–parameterized model and update p^* if needed
    end for
  end for
  return p^*
end procedure

among the most powerful modelling methods in wind energy prediction. The number of possible architectures and the many choices available for them would result in an unmanageable number of model hyperparameters to explore when looking for the best ones. To limit this, we have first fixed some of them to reasonable values that give good results in a first coarse model exploration.

A first such choice is that of the deep architectures to be considered. For deep MLPs we will consider two hidden layers with the same number of units. Our first choice for deep CNNs, which we call standard deep CNN or sdCNN, will have an initial convolutional layer followed by two fully connected layers again with the same number of units. Our second CNN choice, which we call LeNet CNN or lnCNN, will be an adaptation of the well known LeNet–5 architecture [16], that was specifically designed for the MNIST character recognition problem.

We will use the non–symmetric ReLUs at the hidden layers and, as discussed before, for network initialization we will apply the Glorot–Bengio heuristic proposed in [8] of using a 0–symmetric uniform distribution with a width adjusted to the layers' fan-in, scaling then up these initial weights by a factor of 1.5.

The training algorithm we are going to use for all the experiments is conjugate gradient descent (CGD) over random mini–batches. In other words, over each new mini–batch we apply CGD starting at the weights derived over the previous mini–batch; their size clearly affects the performance of the network and we have used sizes of either 200 or 250, i.e., about 6% and 9% of the training sample size. Our error measure is the mean absolute error (MAE)

$$MAE = \frac{1}{N} \sum_{n=1}^{N} |D(x_n; P) - y_n| \,,$$

where $D(x; P)$ denotes the value on pattern x of the current deep network D built using the hyperparameter set P. We use the MAE instead of the more often used squared error as it is the measure of choice in renewable energy, for it represents energy deviation and, thus, the energy to be shed or obtained from other generation sources to compensate errors in wind energy estimates.

As we shall see in the next subsection, the overall MAE evolution during training is decreasing but it often presents spikes due to the use of mini–batches, and this carries on to validation MAE values. In addition, validation MAE seems to stabilize even while training MAE keeps decreasing. Because of this, our model selection strategy is to train a deep NN while there is at least a 1% drop in MAE in the last 100 epochs, with a a maximum of $1,000$ epochs (i.e., goes throught the entire training set). For convolutional networks we will consider each weather variable to define an input feature map; there are thus 8 such features. The above choices leave us with the following hyperparameters to be selected:

- For deep MLPs (which we denote by MLP2) we have to decide on the number (one or two) of hidden layers, the number of hidden units per layer, the weight decay and dropout coefficients, and mini–batch size.
- For the standard deep CNN (which we denote by CNN) we add to the previous deep MLP parameters the convolutional filter and pooling sizes, and their strides.
- For the LeNet CNN (which we denote by LeNet) we also have to decide on the deep MLP parameters but we simplify the other choices by selecting filter and pooling sizes and strides as adequately scaled versions to our problem of the choices made for LeNet–5.

In any case, it is clear that even after the previous simplifications, the number of hyperparameters is too large for an exhaustive grid search. To alleviate this we have used a greedy approach in which we fix first the number of fully connected hidden layers as 2 and then apply Algorithm 1, in which models are evaluated in terms of the MAE over the validation subset. The algorithm performs $n = 50$ external iterations on each of which a concrete hyper–parameter vector p is evaluated. The hyper–parameters considered are the number of hidden units in fully connected layers, the weight decay multipliers used in them, the dropout fraction and the minibatch size. On each external iteration m random choices are made of hyper–parameter indices k and for each a possible updating value p_k is randomly selected from the list $\{v_1^k, \ldots, v_{N_k}^k\}$ of values of the k–th hyper–parameter to be explored. Both random selections are uniform. Actual tested values were

- Hidden unit numbers: 50, 100, 150, 200, 250, 300, 350, 400.
- Weight decay multipliers: 0.1, 0.2, 0.3, 0.4, 0.5.
- Dropout fractions: 0.3, 0.4, 0.5, 0.6, 0.7, 0.8.
- Minibatch size: 50, 100, 150, 200, 250, 300.

For the deep CNNs we fixed the stride to 1 and adjusted filter and pooling sizes by a limited heuristic search; notice that these sizes imply at least four more parameters and a fully random search over the entire parameter set is nearly impossible. The same is true for the number of convolutional feature maps. The just described hyperparameter search results in the following deep NN definitions:

Table 1. Mean Absolute Errors for the Sotavento and REE problems

	MAE Sotavento			MAE REE		
	Test	Validation	Train	Test	Validation	Train
SVR	7.80	6.73	5.62	3.13	3.30	1.01
LeNet	7.63	6.25	5.82	3.13	3.01	2.48
CNN	7.76	6.26	5.39	3.31	3.05	1.96
MLP2	7.76	6.33	5.86	3.37	2.96	1.97
MLP1	8.25	6.41	5.51	3.70	3.10	1.81

- Deep MLPs (MLP2) for Sotavento will have two hidden layers of 250 units, a weight decay coefficient of 0.3 and dropout coefficient of 0.7; mini–batch size is 200. The REE ones will have the same weight decay and dropout coefficients, two hidden layers of 300 units and mini–batch size is 250.
- Standard deep CNNs (CNN) for Sotavento will have a first convolutional layer with 2×6 filters and max pooling is performed over 2×2 patches. This layer is followed by two fully connected layers of 200 units, no weight decay, dropout coefficient of 0.7 and mini–batch size 250. The REE CNN has the same structure; the first layer has now 3×3 filters, max pooling is done over 3×5 patches. This is followed by two fully connected layers of 400 units, weight decay and dropout coefficients are 0.3 and 0.7 respectively and mini–batch size is 200. We used 16 convolutional feature maps for Sotavento and 8 for REE.
- The adapted LeNet–5 (LeNet) network for Sotavento has a first convolutional layer with 2×2 filters and max pooling, and a second one with 4×2 filters and 2×2 max pooling. They are followed by two fully connected 200 unit layers, no weight decay, dropout coefficient of 0.7 and mini–batch size 250. For REE, the LeNet network has a first convolutional layer with 6×8 filters, 2×2 max pooling, and a second one with 6×6 filters and 2×2 max pooling. They are followed by two fully connected 200 unit layers, weight decay and dropout coefficients of 0.3 and 0.7 respectively, and mini–batch size is 200. For both problems we used 16 convolutional feature maps in the first layer and 32 in the second.

3.3 Results

For comparison purposes, we also consider a Gaussian SVR model and a "standard" one–hidden layer, 10–unit MLP. We have used the very well known LIBSVM library [6] and the SVR hyperparameters C, γ and ϵ have been established by a grid search; their optimal values were C=128.0, $\gamma = 3.0518 \times 10^{-5}$ and $\epsilon = 0.0625$ for Sotavento and C=128.0, $\gamma = 12.2078 \times 10^{-5}$ and $\epsilon = 0.01$ for REE. For the standard MLPs we used again Pylearn2–Theano and the optimal parameters were 0.001 weight decay coefficient and 200 mini–batch size in Sotavento and 0.1 weight decay coefficient and 250 mini–batch size for REE.

Table 2. Training complexity parameters and times in seconds for the Sotavento (top) and REE (bottom) deep models

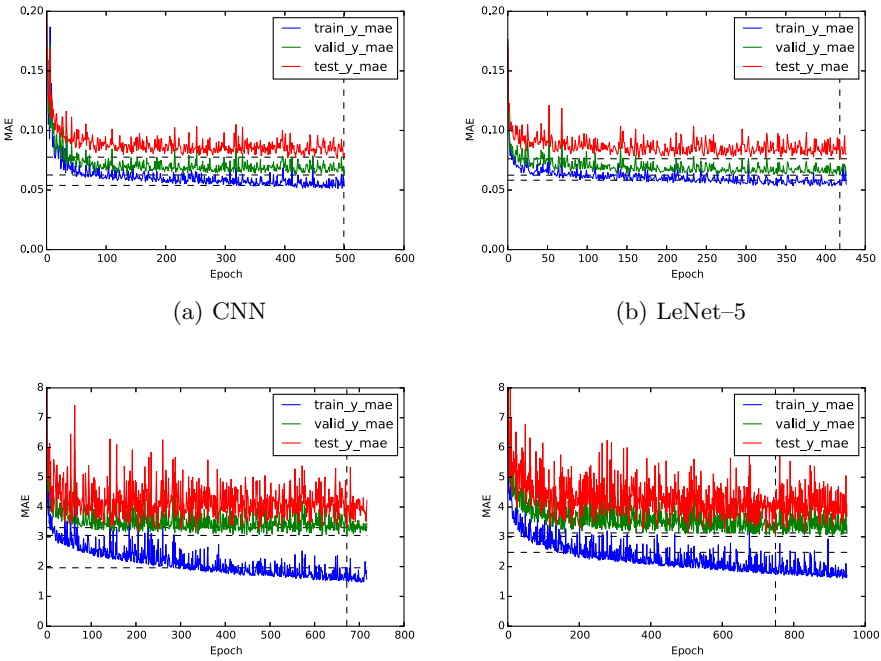
Model	#Params.	#Iters.	Time	Time/Iter.
LeNet	140808	426	1175	2.76
CNN	105736	500	705	1.41
MLP2	332750	259	276	1.07
LeNet	224776	949	19494	20.54
CNN	548176	717	6880	9.60
MLP2	4878300	258	1208	4.68

Table 1 gives training, validation and test errors for the optimal models and the two problems. As it can be seen, the SVR and LeNet–5 models have a similar in performance in the REE problem, followed by the other two deep models; the standard MLP is in a distant last place. However, in Sotavento the LeNet–5 model is clearly the best model while the SVR and the two deep models essentially tie for second place; again, the standard MLP comes in last place. We point out that although we follow a straightforward train–validation–test scheme for model evalution, a more accurate comparison should be made using an appropriate statistical test such as the well known Wilcoxon Rank Sum test, that takes into account not only MAE values but also standard deviations. This requires larger training periods and will be considered in further work.

Figure 1 shows the evolution of the train, test and validation errors for the optimal CNN and LeNet–5 networks for Sotavento (top) and REE (bottom). The large error variations are caused by mini–batch training; while at first sight validation and test error evolution appears smoother for Sotavento, this is partially due to a scale effect (about twice as large for Sotavento than for REE). For Sotavento the smallest errors seem to have essentially reached stable values; this is also the case for the validation and test errors in REE although training error would keep on decreasing, probably because the higher dimensionality of this problem. In both cases the vertical dotted line indicates the epoch with a lowest validation error and the horizontal dotted lines indicate the training, validation and test errors in that epoch. These are the values reported in Table 1.

Finally, in Table 2 we give the give the complexity parameters and training times in seconds for the deep models used in the Sotavento (top) and REE (bottom) problems. As it can be seen, all models are rather large, and more so those used for REE (remember that input dimensions are respectively 1,080 and 15,960). Besides, while the convolutional networks have less parameters than MLP2, their feedforward passes are much costlier due to the convolution operations and the same is true for the backpropagation of gradients. We observe that the smaller number of weights in LeNet for REE is due to the larger filters used.

It follows that deep training is rather costly and must take advantage of all possible hardware–based improvements available. In our case experiments



(a) CNN (b) LeNet–5

(c) CNN (d) LeNet–5

Fig. 1. Training, validation and test evolution of the optimal CNN and LeNet–5 networks for Sotavento (top) and REE (bottom)

have been run on a machine equipped with a NVidia Tesla K40 GPU and the Pylearn2–Theano framework. Working with Pylearn2 eases somewhat the development process, since most of it is written in Python.

There are other platform alternatives with the already mentioned Caffe being an interesting one, as its core is written in C++ and CUDA, which should result in a performance improvement. Another important improvement comes from the NVidia cuDNN library, that inter–operates perfectly with Pylearn2–Theano (more than doubling the performance of the previous version) and Caffe.

4 Conclusions

While undeniably very powerful, the optimal architectures and best hyper–parameters of deep neural networks are also quite hard to set up and select. However, when properly tuned, they can often produce better results than other classical models, as we have demonstrated here on two wind energy problems. The use of weather variables gives to both problems a bi–dimensional input structure; moreover, these variables can be naturally seen as input channels. This may suggest a reason why the best deep results were obtained using convolutional

layers. Deep network training is also very computationally demanding but, on the other hand, lends itself extremely well to the use of GPUs and the large speed-ups that they allow.

In any case, the work presented here has to be considered as a first step. A first line of further work is to consider other convolutional architectures, specially of the AlexNet type ([14]). Another natural option is to try to reduce variance by combining several deep models (notice that they have naturally a low bias). The usual choice in standard MLPs is to repeat training from different random initializations but given the high validation variability during training, a simpler, less costly possibility is to select a certain number M of the models with smallest validation that were obtained in a single training run as the ones followed here.

Furthermore, the tremendous activity in deep learning is producing a large number of proposals for network initialization and architectures as well as model training and regularization. We are also pursuing some of these options.

Acknowledgments. With partial support from Spain's grants TIN2013-42351-P (MINECO) and S2013/ICE-2845 CASI-CAM-CM (Comunidad de Madrid), and the UAM–ADIC Chair for Data Science and Machine Learning.

The second author is also kindly supported by the FPU-MEC grant AP-2012-5163. The authors gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at UAM and thank Red Eléctrica de España for kindly supplying wind energy production data.

References

1. European center for medium-range weather forecasts. http://www.ecmwf.int/
2. Global forecast system. http://www.emc.ncep.noaa.gov/index.php?branch=gfs
3. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I.J., Bergeron, A., Bouchard, N., Bengio, Y.: Theano: new features and speed improvements. In: Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop (2012)
4. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems 19 (NIPS 2006), pp. 153–160 (2007). http://www.iro.umontreal.ca/~lisa/pointeurs/BengioNips2006All.pdf
5. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010. Oral Presentation
6. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**(3), 27:1–27:27 (2011). software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
7. Ciresan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011, Barcelona, Catalonia, Spain, July 16–22, 2011, pp. 1237–1242 (2011). http://ijcai.org/papers11/Papers/IJCAI11-210.pdf

8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010), vol. 9, pp. 249–256, May 2010
9. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011), April 2011
10. Goodfellow, I.J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., Bengio, Y.: Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214 (2013). http://arxiv.org/abs/1308.4214
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR abs/1502.01852 (2015). http://arxiv.org/abs/1502.01852
12. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006). http://www.sciencemag.org/content/313/5786/504.abstract
13. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. CoRR abs/1408.5093 (2014). http://arxiv.org/abs/1408.5093
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
15. Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodriguez-Sanchez, A., Wiskott, L.: Deep hierarchies in the primate visual cortex: What can we learn for computer vision? IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1847–1871 (2013)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
17. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient BackProp. In: Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 1524, pp. 9–50. Springer, Heidelberg (1998)
18. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014). http://jmlr.org/papers/v15/srivastava14a.html
19. Sutskever, I., Martens, J., Dahl, G.E., Hinton, G.E.: On the importance of initialization and momentum in deep learning. In: Dasgupta, S., Mcallester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning (ICML 2013), vol. 28, pp. 1139–1147. JMLR Workshop and Conference Proceedings, May 2013. http://jmlr.org/proceedings/papers/v28/sutskever13.pdf

Ensemble of Classifiers for Length of Stay Prediction in Colorectal Cancer

Ruxandra Stoean[1(✉)], Catalin Stoean[1],
Adrian Sandita[1], Daniela Ciobanu[2], and Cristian Mesina[3]

[1] Department of Computer Science, Faculty of Mathematics and Natural Sciences, University of Craiova, Craiova, Romania
{ruxandra.stoean,catalin.stoean}@inf.ucv.ro, asandita@gmail.com
[2] Department of Internal Medicine, Emergency County Hospital Craiova and University of Medicine and Pharmacy of Craiova, Craiova, Romania
elada192@yahoo.com
[3] Department of Surgery, Emergency County Hospital Craiova and University of Medicine and Pharmacy of Craiova, Craiova, Romania
mesina.cristian@doctor.com

Abstract. The paper puts forward an ensemble of state-of-the-art classifiers – support vector machines, neural networks and decision trees – to estimate the length of stay after surgery in patients diagnosed with colorectal cancer. The three paradigms are brought together in order to achieve both a more accurate prediction through a voting scheme and transparency of the discriminative guidelines through visual rules. The results support the theoretical assumptions and are confirmed by the physicians.

Keywords: Classification · Length of stay · Colorectal cancer · Ensemble of methods · Prediction accuracy · Decision guidelines

1 Introduction

Medicine has been advancing to the point where it is highly possible to cure cancer provided a timely detection, appropriate surgical intervention and an adequate treatment. Even so, hospitals would be able to sustain better medical care if the length of stay (LOS) of patients undertaking surgery were accurately forecasted, based on previous records referring imaging and biopsy indicators.

The aim of the paper is then to develop an intelligent decision (ID) framework for hospital management and practitioners in estimating the LOS of patients diagnosed with colorectal cancer, a type of increasing incidence. Whereas a significant enhancement in cure chances has been achieved through early diagnosis and corresponding surgery [4], [15], [17], the information on the expected hospitalization period is beneficial for an optimal allocation of beds and other medical resources for the further treatment of these patients.

Given the available data whose relationship trigger the hospitalization time and whose amount is sufficiently high to be hardly interpreted by the human

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 444–457, 2015.
DOI: 10.1007/978-3-319-19258-1_37

experts, the estimation of LOS for patients diagnosed with colorectal cancer is thus planned to be extracted by means of state-of-the-art ID approaches. The real-world data collection is provided by the University of Medicine and Pharmacy of Craiova, Romania, and is made available for future studies (see next section). It contains pre-operative medical information from clinical and paraclinical imaging exploration, post-operative attributes resulting from the microscopic examination of the pieces resulting after excision, as well as the cancer stage.

The paper therefore aims to establish the optimal (combination of) ID methods for an accurate hospital LOS prediction in patients operated for colon cancer and for a simultaneous provision of the guidelines leading to the established decision.

2 Materials

The data set comes from the University of Medicine and Pharmacy of Craiova, Romania, and concerns the hospitalization period of 368 patients diagnosed with colorectal cancer that underwent surgery between 2012 and 2014.

The set encompasses pre-operative information, which was obtained from computed tomography, colonoscopy with punch biopsy, irigography and echography, and post-operative indicators. The cancer stage was established after sampling tissues obtained after various categories of surgery. The pieces underwent macroscopic examination and microscopic analysis to evaluate the type of tumor, aggressiveness, grading, the extent of invasion of the colon tunicae, the presence of metastases and the status of the satellite lymphatic ganglions of the tumor.

To summarize, the collection specifically comprises of the 8 attributes that the protocol requires, namely socio-demographic variables (age, sex) and clinical information (operation type, tumor topography, stage, tumor parameter (T), adenopathy parameter (N) and metastases (M)).

Hospital stay after surgery is usually weekwise, so the dependent variable – the hospitalization period – was provided as split into three intervals: 1-7 days (one week) – short stay (177 cases), 8-14 days (two weeks) – medium stay (96 cases), over 14 days – long stay (95 cases). LOS is thus referred in the data set as an ordered dependent variable with the three stated categories.

An overview of the independent variables – the socio-demographic and clinical attributes – is given in Table 1, while the data set is available for download for future reference at the web page https://sites.google.com/site/imediatreat/. Under the *Surgery* type, rectal biopsy is a method for diagnosis and not for therapy. Some patients with rectal cancer had a rectal biopsy performed – a fragment from the tumor had been sent for anatomopathological examination. The whole tumor can also be sampled if it is low situated – then the method is therapeutical, provided that the tumor invades only the mucosa.

Table 1. Indicators and definition domains for the colon cancer data set

Variable	Range
Age	[25, 92]
Sex	F/M
Topography	cecum ascending colon descending colon sigmoid colon transverse colon rectosigmoid junction rectum hepatic flexure splenic flexure
Stage	1 (the tumor invades the mucosa and the submucosa) 2 (it invades the tunica muscularis) 3 (it invades all colon tunicae, without exceeding the serosa) 4 (it invades the neighboring organs and gives metastases at a distance)
T	T0 - carcinoma in situ, localized only at mucosa (not present in the data set) T1 - mucosa and submucosa T2 - tunica muscularis T3 - serosa T4 - invasion at neighboring organs Tx - undetermined at the time (not present in the data set)
N	N0 - no adenopathy (not present in the data set) N1 - invasion to up to three ganglions N2 - invasion to more than three ganglions Nx - undetermined at the time
M	M0 - without metastases (not present in the data set) M1 - metastases at a distance M1 hep - M1 hepatic Mx - undetermined at the time
Surgery	Ileotransverse-anastomosis Raybard colectomy Ileosigmoid anastomosis Segmental colectomy Left hemicolectomy Subtotal colectomy Abdominoperineal rectal amputation Hartmann rectosigmoidian resection Rectal biopsy Rectosigmoidian resection Loop colostomy Total colectomy Right hemicolectomy Ileostomy Low anterior rectosigmoidian resection (Dixon procedure)

3 State of the Art

The problem of LOS estimation for different diseases by ID means has been extensively studied in recent literature [1], [2]. In particular for the current problem, the number of colorectal carcinoma cases is increasing worldwide and so are the related costs for its diagnosis, grading, surgery and treatment. An accurate approximation for the LOS triggers efficient planning and better patient care, as argued by all current studies. The literature survey revealed, as a rule of thumb, the use of logistic regression (LR) for LOS prognosis in colorectal cancer hospitalization.

Accordingly, paper [12] performs a statistical analysis of the tendency of LOS and LR identifies the influential factors as age, marital status and higher levels of co-morbidities. In [16], age, source, surgical type, tumor size, tumor site and tumor stage determined an extended LOS, as discovered again by LR. In [5], distal (vs proximal) bowel resection, benign pathology, open technique, increasing age, co-morbidity and social deprivation were associated to prolonged LOS once more by the same methodology. Coronary artery disease and postoperative complications were found by study [14] as correlated to longer LOS.

4 Methodology

Since the medical problem is formulated as a classification task, any particular ID technique initially learns the association between the independent variables and the acknowledged outcome in the training phase. The derived model subsequently makes a prediction on the probable LOS category of the new examples, in accordance to the learnt patterns. Depending on the selected methodology, the output may consist of solely the prediction accuracy or accompanying explanations.

4.1 Preamble

The choice of methods to solve the task was therefore influenced by three factors: (1) select an approach that has shown high accuracy in practice, (2) pick a technique that also provides understandable guidelines and (3) establish an efficient means to balance good accuracy and comprehensible decision guidelines for the medical staff.

When it comes to the main actors in the machine learning arena, support vector machines (SVM) and neural networks (NN) are those with the leading role for an accurate classification. However, they both lack transparency of the decision process and comprehensibility of the resulting verdict [10], which are compulsory for a field like medicine, where vital conclusions are derived on the base of a correct but also trustworthy support [7], [11]. There is also the class of rule-centered classifiers, like decision trees (DT), where the logic of discrimination can be easily followed from a tree structure, where nodes hold the discriminative attribute thresholds and their descending branches eventually lead to the

outcomes lying in the leaves. Their prediction is generally accurate, although slightly lower than that of the SVMs and NNs.

Therefore, the three well-known ID paradigms are addressed in turn in the attempt to solve the colorectal cancer LOS problem. It is finally aimed to encompass both the prediction strength of the SVM/NN approaches and the visual understandability of the decision guidelines following DT. Ensemble learning is therefore addressed such that all techniques learn from the same data collection and make a unified prediction on the unknown cases. Since the existing literature entries for targeting this problem use mainly LR, the ordered Logit model for (more than two) ordered categories for the dependent variable is also included in the study.

Before detailing each approach in turn, let the classification problem be formulated as consisting of m pairs $(\mathbf{x}_i, y_i)$, where each sample $\mathbf{x}_i$ holds the values for the n predictive attributes and $y_i \in \{1, ..., k\}$ is the response, given there are k categories of the task.

4.2 Logistic Regression

The employment of the LR model in the computational literature related to the LOS problem is justified by its somewhat similarity in form to the multiple regression technique. The method makes a forecast on the value of a converted dependent variable by what is called the logit transformation [6].

While the traditional LR considers only binary classification problems, its extension referred to as the ordered logit model (or proportional odds model) treats tasks with multiple classes with the supplementary knowledge of their order (and consequent relationship).

Given any training sample of attributes $\mathbf{x}$ and output y, if p_j denotes the proportion of data labeled with class j of the problem, then cumulative logits are formed (1), where $j = 1, ..., k$ [3].

$$\ln\left(\frac{Pr(y \leq j)}{Pr(y > j)}\right) = \ln\left(\frac{p_1 + ... + p_j}{p_{j+1} + ... + p_k}\right) \tag{1}$$

The transformed output is then modeled as a linear combination of the predictive variables (2), where α_j, $j = 1, ..., k$, is the intercept and $\mathbf{w}$ denotes the vector of coefficients for the attributes.

$$\ln\left(\frac{Pr(y \leq j)}{Pr(y > j)}\right) = \alpha_j + \mathbf{x} \cdot \mathbf{w} \tag{2}$$

Estimation is then conducted by maximum likelihood (Newton-Raphson algorithm in the current implementation). Once the model is discovered, the probabilities of each category for a new sample $\mathbf{x}^*$ are derived from the computations in (3) and (4) [3].

$$Pr(y^* = 1|\mathbf{x}^*) = Pr(y^* \leq 1|\mathbf{x}^*) = \frac{\exp(\alpha_1 + \mathbf{x}^* \cdot \mathbf{w})}{1 + \exp(\alpha_1 + \mathbf{x}^* \cdot \mathbf{w})} \tag{3}$$

$$Pr(y^* = j|\mathbf{x}^*) = Pr(y^* \leq j|\mathbf{x}^*) - Pr(y^* \leq j - 1|\mathbf{x}^*), j = 2, ..., k \tag{4}$$

4.3 Support Vector Machines

SVM model the division between classes by means of a hyperplane of coefficients $\mathbf{w}$ and b that has minimal training error and generalizes well on new data (with a maximal margin of separation) [18]. Minimal deviations to the output ξ_i, $i = 1, 2, ..., m$, are allowed and penalized by a parameter customarily denoted by C. If the samples cannot be linearly discriminated, then a nonlinear surface is obtained through the kernel trick, which implies that data are mapped into a higher dimensional space where a linear surface performs the separation. Kernels are commonly taken in either a polynomial expression or Gaussian (radial) formulation.

The reached optimization problem (5) is solved by an extension of the Lagrange multipliers technique. A dual formulation is reached and the optimal Lagrange multipliers are considered as the solutions of the system resulting by setting the gradient of the new objective function to zero. Once the Lagrange multipliers are known, the output for every new data sample can be also computed and provided to the user.

$$\begin{cases} \text{find } \mathbf{w}, b \text{ such as to minimize the objective } \|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \xi_i, C > 0 \\ \text{subject to the constraints } y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, ..., m \ . \end{cases} \tag{5}$$

While SVM is also primarily used for binary classification, it has been as well extended to handle multiple categories of a problem through two approaches: one-against-all and one-against-one [9]. In the following, only the latter is summarized because it is employed in the current implementation. The approach builds $\frac{k(k-1)}{2}$ classifiers, where k is the number of classes. Each i^{th} SVM is trained on data from every two categories in turn, i and j, where data of class i are positive and those of class j are negative. Once all the corresponding hyperplanes are determined, the label for a new sample is determined through a vote. For every built SVM, the derived class for that example receives a vote; in the end, the class with the highest number of received votes wins to label the sample.

4.4 Neural Networks

NN [8] conduct automatic learning by simulating the biological neural interactions of the human brain. Simplistically, the neurons send information from one to another and the intensity of the synaptic connection between them influences the transmission. Analogously, units of a NN spread knowledge through the network and the synaptic weighting determines the performance of the artificial learning.

The outside (input) units of the NN take the information from the environment (problem) which is processed by the interconnected hidden neurons (organized in one or more layers) and the network eventually produces its response (output) to that given information. Learning is achieved by adjusting the synaptic weights in concordance to the environment.

Each neuron u can be described by (6), where **in** is its input, **w** is the vector of synaptic weights and b is the bias.

$$u = \mathbf{w} \cdot \mathbf{in} + b \tag{6}$$

Then, its output is given by (7), where φ is the activation function.

$$out = \varphi(u) \tag{7}$$

While there are many NN architectures, the current implementation considers the feedforward type with one hidden layer. The information is propagated from the input layer (referring each sample $\mathbf{x}_i$ in turn) to the output through the hidden units only in a forward fashion. Learning is supervised, i.e. the weights are optimized on the base of the difference between the actual target y_i of each sample and the NN forecasted one y_i^*. The initial weights are randomly selected from a chosen interval. The number of units in the hidden layer is a parameter of the approach. The activation function is selected as the widely used sigmoid formulation, of which one example is the logistic function.

As for SVM before, from the practical perspective, the learning process takes place in a black box, which receives the input and offers its predicted output to the user with no further explanations of the decision making process.

4.5 Decision Trees

Highly appreciated because they give the possibility of an immediate and understandable visual guideline of the learning model, DT are the last paradigm considered in the study.

A tree consists of internal nodes, where each of them represents a test for an attribute of the problem and divides the objects on the base of the condition it holds. Every branch stands for the partition resulting from the test. Each leaf node denotes the reached class. The data samples are recursively partitioned until leaf nodes of the same (or dominant) category are obtained.

In order to have an efficient partitioning of the data at a node, one of several splitting criteria can be utilized. Strictly referring to the two choices allowed by the current implementation, one is the Gini index (which must be minimized such as only one class is present at the node) and the maximal information gain (or the minimal entropy or impurity of the node).

For the DT induction, the current implementation employs the classification and regression tree (CART) approach. As regards the prediction indicators for every class [6], the prior probabilities are chosen as proportional to the number of objects in that class and the misclassification costs are all selected as equal to one, as no category needs to be more attentively predicted than the others.

Once the DT is constructed, rules must be extracted from the model. They are naturally of the if-then type and are formed by descending on the branches and building the condition part as conjunctions between the tests of the internal nodes. The conclusions of the rules are established once leaf nodes are reached.

4.6 Ensemble Construction

In order to have maximal benefits from both the discriminative power of SVM/NN and the illustrative possibilities of the DT, once all three classifiers are trained, a voting mechanism is used to establish the class for fresh examples. For every new case, one label is derived from each method and the class most frequently appearing in this triple response is assigned as the combined vote result. If the three predictions are however all distinct, then priority is given to the decision of the SVM (as pre-experimentally proven most accurate).

5 Experimental Results

Let it be recalled that the pool of candidates for the experimental stage comprises of SVM, NN, DT, Logit and a combination that uses the first three to vote for the class. Although the results are close with respect to accuracy, at least for the first three, there are some particularities that will be further detailed in the current section.

5.1 Pre-experimental Planning

While making some preliminary runs of the tested methods, we noticed that the confusion matrices are considerably different from each other. As the misclassified cases are thus not the same within all approaches, it was decided to gather the top performers and use all of them to vote for each new sample to be classified, since their combination may lead to a complementary learning and obtain more accurate predictions.

5.2 Task

The aim of the experiments is to search for a machine learning approach that can provide an acceptable accuracy for the considered three-class data set. Additionally, a rule (or more) to act behind the decision engine and explain the mechanics is desirable and highly appreciated by the practitioners.

5.3 Setup

Four different methodologies are considered for the classification task: SVM, DT, NN and Logit. While the first three are the top classifiers that are widely used in real-world scenarios, the latter is considered because most of the LOS problem references employ LR.

The distribution of cases among classes in the amounts 177-96-95 (short-medium-long stays) is balanced, because it is typical that a larger part of the patients have a short hospital stay following the types of surgery related to colorectal carcinoma.

In each run of an algorithm, the data is randomly split into 2/3 training and 1/3 test disjoint sets. Prediction accuracy is calculated as the percentage of correctly labeled test cases. Each method is repeated 30 times and the average value and standard deviation of the accuracies are reported. Additionally, for a deeper understanding of the mislabelings, confusion matrices are computed also as an average over the 30 repeated runs. In order to make the comparison between the results of the various approaches more objective, when a split between training and test sets is generated, all techniques learn from the same training data and are applied on the same test set.

The R software is used to manage the implementations for the classifiers: the used packages are *e*1071 for SVM, *rpart* for DT, *nnet* for NN and *ordinal* for Logit (the *clm* function). In general, the default values for the parameters of the approaches are used. For Logit, various combination of values had been tried, but no significant improvements were noticed, so the default values were kept as well. The only exception was the NN, which was found to be sensitive to changes and, after some manual tuning, we set the number of units in the hidden layer to 10, random weights in the [-0.1, 0.1] interval, weight decay to 10^{-4} and the maximum number of iterations to 100. Since the output is a factor, both the SVM and the DT packages implicitly perform classification. The DT learning methodology is chosen as the one based on information gain, due to the categorical nature of the attributes. For SVM, a linear hyperplane performed best according to pre-experimental tests. Finally, penalization for violating the SVM constraints has the default value $C = 1$.

The SVM, DT and NN are used to provide the output for the samples in the test set through simple voting as explained in subsection 4.6.

5.4 Results and Visualization

Table 2 contains the average prediction accuracies, standard deviations and confusion matrices for the four tested methods from 30 runs.

In order to compare their prediction accuracies, we applied the Wilcoxon rank-sum nonparametric test. The Logit and every other method show a statistically significant performance difference (with the former being worse). For each of the other pairs of approaches, the null hypothesis (which states that every two approaches do not perform differently from each other as derived from the test), cannot be rejected because the obtained p-values were higher than the significance level of 0.05. The closest results to rejection are those for the voting combination as compared to the NN (p-value of 0.08) and for SVM vs NN (p-value of 0.13). Yet, the combination still yields a compact ID framework, from which the specialist can eventually evaluate all the resulting information and establish the final verdict.

Besides the Wilcoxon rank-sum test, the extent of agreement between the labeling dictated by every two methods (raters) is computed via Cohen's kappa statistics. Its results are shown in Table 3.

The rule set behind one random run of the DT is illustrated in Fig. 1, while the rule that correctly labels most of the samples as *short term stay* is described

Table 2. Mean resulting accuracies and confusion matrices (true on the columns and predicted on the rows) for 30 repeats with random cross validation

Method	Accuracy and st. dev.		True		
		Predicted	Short	Medium	Long
SVM	73 ± 3.9	Short	51.63	8.2	7.93
		Medium	3.83	21.17	5.43
		Long	4.03	3.5	16.27
DT	72.32 ± 4.22	Short	51.5	7.43	7.8
		Medium	3.43	19.13	4.23
		Long	4.57	6.3	17.6
NN	71.2 ± 4.26	Short	48.7	7.93	7.33
		Medium	5.13	20.33	4.47
		Long	5.67	4.6	17.83
Logit	60 ± 4.6	Short	51.07	16.1	8.83
		Medium	4.2	6.4	5
		Long	4.23	10.27	15.6
Vote	73.14 ± 4.37	Short	51.63	7.63	8.63
		Medium	3.97	21.7	5.1
		Long	3.9	3.53	15.9

in detail in Fig. 2. To identify the meaning of the letters from every node, the order of the values for each indicator from Table 1 should be inspected. It actually coincides with the alphabetical order of the levels for each indicator as they appear in the data set and the order of the respective encoding is preserved in Table 1.

In order to more explicitly show the sequence of application of the obtained rules for a single test case, two samples are randomly chosen from the data set. They are v_1 : $(age = 76, sex = F, topography = cecum, stage = 3, T = T3, N = Nx, M = Mx, surgery = 3)$ and $v_2 : (age = 79, sex = F, topography = cecum, stage = 2, T = T2, N = Nx, M = Mx, surgery = 3)$. For establishing a label for v_1 by the tree in Fig. 1, the first test asks whether T is equal to $T1$ or $T2$ (denoted by a and b); as $T = T3$, the right branch is followed to the second test that verifies the type of *surgery*. Since in this case the *surgery* type is the third in the table (i.e. corresponding to the third letter of the alphabet c), then the left branch directly leads to a leaf node denoting the *medium* category, which is in fact identical to the real outcome for v_1. Analogously for v_2, the test at the root node leads the sample to the left branch ($T = T2$), where *topography* is verified. For this example, *topography* = *cecum* (whose position in the ordering from Table 1 corresponds to the first letter a in the tree encoding for attribute values) and, as such, the left branch is followed again. The value for *surgery* is then tested and, since this is equal to the third type as given in Table 1, the sample descends to the next node through the right branch, where *surgery* types were chosen to discriminate again among objects. Since the value is once more c (third surgery type = third letter of the alphabet), the right branch is followed and leads to the leaf node denoted by *long*, which is also the actual label of v_2.

Table 3. Cohen's kappa statistics. A small value for the kappa variable means that the agreement is low (kappa = 0 means it is same as random), while a value close to 1 means almost perfect agreement.

	SVM	DT	NN	Vote
Original outputs	0.56	0.55	0.54	0.56
SVM	-	0.79	0.73	0.94
DT	-	-	0.69	0.85
NN	-	-	-	0.72

5.5 Discussion

The Logit method provides very poor results as compared to the other three algorithms and we will not consider it in the rest of the discussion subsection.

As concerns the rest, SVM, DT, NN and Vote, both the prediction accuracy and standard deviation are very similar (Table 2), also demonstrated by statistical testing.

The confusion matrices show that there are misclassified patients from short to long or vice-versa for all approaches. The cause is that there are samples with almost similar values that stayed for very different periods of time in the hospital. The physicians explain that such differences appear because of various complications that could not be included in the data, like the presence of anemia, diabetes, combination with other diseases etc. Information regarding the glycemia or the level of iron-deficiency could not be put in the data set because they were not measured for all patients, but only for those where there existed certain suspicions.

Table 3 contains the level of agreement on the labeling (again over 30 repeats) between every two approaches. Additionally, the original sample outputs in the test set are verified against the results obtained by each method. There does not exist a clear unanimously accepted interpretation for the level of agreements, however very often their meaning is taken as the one suggested in [13]: negative values indicate no agreement, 0 to 0.20 a slight one, 0.21 to 0.40 a fair agreement, 0.41 to 0.60 denote a moderate one, 0.61 to 0.80 a substantial one, and 0.81 to 1 an almost perfect agreement. It is interesting to observe in the last column of the table the level of agreement between Vote and the other approaches: most of the labelings are thus taken from SVM, then from DT and finally, from NN. The difference between the agreement of the original outputs vs. SVM and that for original vs. Vote appears only on the third decimal, so it is not visible in the table.

From Fig. 1 it can be seen that the lowest entropy splitting index is obtained for attribute T equaling $T1$ or $T2$. The most important branch of the tree is the one that goes to the left, taking into account attributes T, $Topography$ and $Surgery$, as the number of correctly classified patients on the leaf with the label *short* is 70; it is separately illustrated in Fig. 2. The next important part of the rule set is the one that goes also to the left from T (*T1* or *T2*) and then

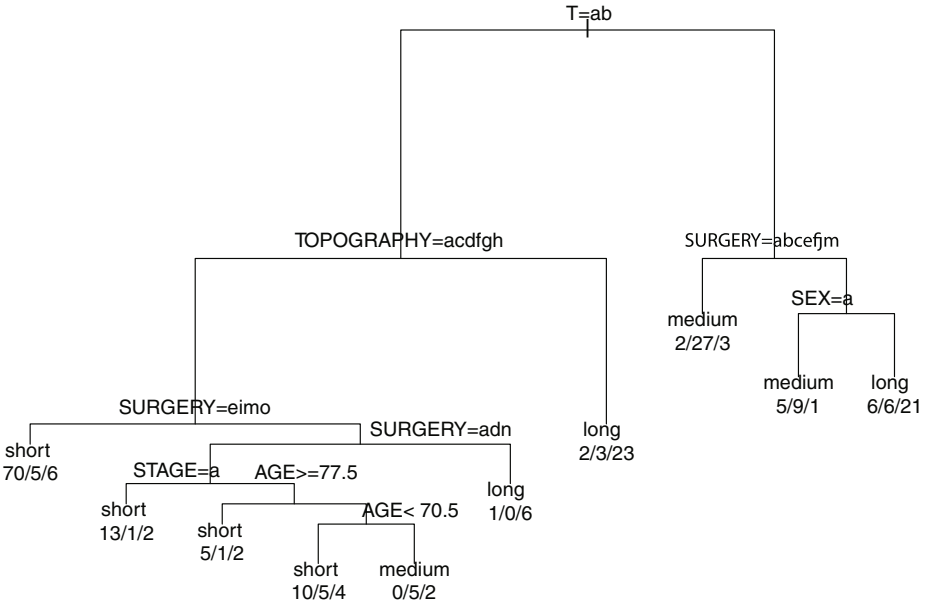


Fig. 1. Decision rules provided by the DT for classifying the test samples. The letters from each node of an ordinal attribute signify the possible values (they are considered in the alphabetical order of the encoding in the data set) for following the left branch. The numbers from each leaf show how the test samples are labeled as short/medium/long stays, providing thus additional information about the *purity* of this branch.

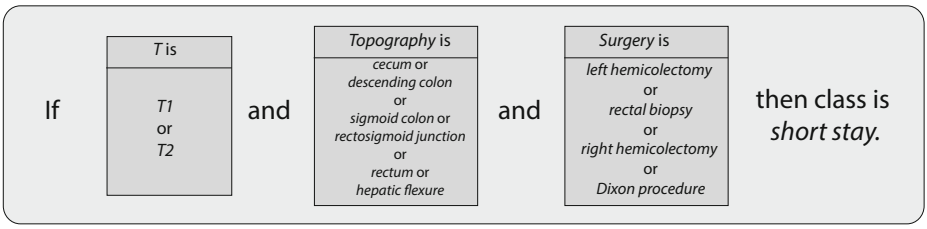


Fig. 2. A rule provided by the DT for accurately classifying most of the test samples as *short term stay*

decides on *Topography* as *ascending colon*, *transverse colon* or *splenic flexure*, conducting to *long term care* and 23 correctly classified patients.

6 Conclusions

From the practical perspective, the framework achieved the creation of a resource saving ID support for medical specialists dealing with persons operated for colorectal carcinoma, with the final purpose of better quality patient care.

Four machine learning approaches are tested on the LOS data set, i.e. SVM, DT, NN and Logit. A simple voting scheme is applied to combine SVM, DT and NN for a slightly more reliable prediction accuracy. The rules put forward by the DT are acknowledged with interest by the physicians, as valuable discriminative details are revealed by the model.

In the near future, we plan to study several means to measure the importance of each indicator and also to boost performance through bootstrap aggregation and the inclusion of other machine learning algorithms.

Acknowledgments. The authors would like to thank the reviewers for their helpful comments and acknowledge the support of the research grant no. 26/2014, code PN-II-PT-PCCA-2013-4-1153, entitled IMEDIATREAT - Intelligent Medical Information System for the Diagnosis and Monitoring of the Treatment of Patients with Colorectal Neoplasm - financed by the Romanian Ministry of National Education (MEN) - Research and the Executive Agency for Higher Education Research Development and Innovation Funding (UEFISCDI).

References

1. Belciug, S., Gorunescu, F.: Improving hospital bed occupancy and resource utilization through queuing modeling and evolutionary computation. Journal of Biomedical Informatics **53**(0), 261–269 (2015)
2. Bhattacharjee, P., Ray, P.K.: Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: A review and reflections. Computers and Industrial Engineering **78**(0), 299–312 (2014)
3. Breheny, P.: Proportional odds models. http://web.as.uky.edu/statistics/users/pbreheny/760/S13/notes/4-23.pdf (accessed: 2015–04-02)
4. Ciobanu, D., Mesina, C., Streba, L., Gruia, C.L., Ditescu, D., Sarla, C.G., Petrescu, F.: The role of immunohistochemistry in diagnosing a synchronous colon tumor. Romanian Journal of Morphology and Embryology **55**(3 suppl), 1209–1213 (2014)
5. Faiz, O., Haji, A., Burns, E., Bottle, A., Kennedy, R., Aylin, P.: Hospital stay amongst patients undergoing major elective colorectal surgery: predicting prolonged stay and readmissions in nhs hospitals. Colorectal Disease **13**(7), 816–822 (2011)
6. Gorunescu, F.: Data Mining - Concepts, Models and Techniques, Intelligent Systems Reference Library, vol. 12. Springer (2011)
7. Gorunescu, F., Belciug, S.: Evolutionary strategy to develop learning-based decision systems. application to breast cancer and liver fibrosis stadialization. Journal of Biomedical Informatics **49**(0), 112–118 (2014)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, 2nd edn. (1998)
9. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks **13**(2), 415–425 (2004)
10. Huysmans, J., Baesens, B., Vanthienen, J.: Using rule extraction to improve the comprehensibility of predictive models. Tech. rep., KU Leuven (2006)
11. Iancu, I.: Intuitionistic fuzzy similarity measures based on frank t-norms family. Pattern Recognition Letters **42**, 128–136 (2014)

12. Kelly, M., Sharp, L., Dwane, F., Kelleher, T., Comber, H.: Factors predicting hospital length-of-stay and readmission after colorectal resection: a population-based study of elective and emergency admissions. BMC Health Services Research **12**(1), 77 (2012)
13. Landis, J., Koch, G.: The measurement of observer agreement for categorical data. Biometrics **33**(1), 159–174 (1977)
14. Leung, A., Gibbons, R., Vu, H.: Predictors of length of stay following colorectal resection for neoplasms in 183 veterans affairs patients. World Journal of Surgery **33**(10), 2183–2188 (2009)
15. Mesina, C., Vasile, I., Ciobanu, D., Calota, F., Gruia, C.L., Streba, L., Mogoanta, S.S., Parvanescu, H., Georgescu, C.V., Tarnita, D.N.: Collision tumor of recto-sigmoidian junction - case presentation. Romanian Journal of Morphology and Embryology **55**(2 suppl), 643–647 (2014)
16. Perng, D.S., Lu, I.C., Shi, H.Y., Lin, C.W., Liu, K.W., Su, Y.F., Lee, K.T.: Incidence trends and predictors for cost and average lengths of stay in colorectal cancer surgery. World Journal of Gastroenterology **20**(2), 532–538 (2014)
17. Richter, J.M., Campbell, E.J., Chung, D.C.: Interval colorectal cancer after colonoscopy. Clinical Colorectal Cancer **14**(1), 46–51 (2015)
18. Vapnik, V.: Statistical Learning Theory. Wiley (1998)

Interactive and Cognitive Environments

Monitoring Motor Fluctuations in Parkinson's Disease Using a Waist-Worn Inertial Sensor

Carlos Pérez-López[1(✉)], Albert Samà[1], Daniel Rodríguez-Martín[1], Andreu Català[1], Joan Cabestany[1], Eva de Mingo[1], and Alejandro Rodríguez-Molinero[2]

[1] Technical Research Centre for Dependency Care and Autonomous Living (CETpD), Universitat Politècnica de Catalunya – BarcelonaTech (UPC), Barcelona, Spain
{carlos.perez-lopez,albert.sama,daniel.rodriguez-martin, andreu.catala,joan.cabestany}@upc.edu, itzahe1@hotmail.com
[2] Clinical Research Unit, Fundación Sant Antoni Abat (Consorci Sanitari del Garraf), Barcelona, Spain
Rodriguez.molinero@gmail.com

Abstract. Parkinson's disease (PD) is the second most common neurodegenerative disorder. First appreciable symptoms in PD are those related to an altered movement control. Current PD treatments temporally revert the symptoms, but they do not prevent disease's progression. At the beginning of the treatment, the antiparkinsonian effect of the medication is very evident and symptoms may completely disappear for hours; however, as disease progresses, motor fluctuations appear. Collecting precise information on the temporal course of fluctuations is essential for tailoring an optimal therapy in PD patients and is one of the main parameters in clinical trials. This paper presents an algorithm for wearable devices to automatically detect patient's motor fluctuations based on inertial sensors. The algorithm has been evaluated in 7 PD patients at their homes without supervision and performing their usual activities. Results are a mean sensitivity of 99.9% and a mean specificity of 99.9%.

Keywords: Inertial sensors · Parkinson's disease · Motor fluctuations · Ambulatory monitoring

1 Introduction

Parkinson's disease (PD) is the second most common neurodegenerative disorder after Alzheimer's disease. The pathology of the disease is characterized by a protein accumulation into brain neurons inclusions, called Lewy bodies, and the insufficient activity in the neurons that produce dopamine, one of the main neurotransmitters involved in the control of movement, in the substantia nigra, which is located within midbrain [1].

First appreciable symptoms in PD are those related to an altered movement control, such as tremor, rigidity or bradykinesia (slowness of movement). These symptoms are caused by a decrease in dopamine levels, due to the degeneration of the nerve cells in the brain that produce it. Current PD treatments are based on increasing its production with dopamine agonists, among which, levodopa is the most used one. These

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 461–474, 2015.
DOI: 10.1007/978-3-319-19258-1_38

treatments temporally revert the symptoms, but they do not prevent disease's progression. At the beginning of the treatment, dopaminergic effect is very evident and symptoms may completely disappear for hours; however, as disease progresses, motor fluctuations appear. These motor fluctuations are presented as an alternation between ON and OFF periods. During ON periods, patients report that they feel relatively clear and in control of their movements. Furthermore, PD symptoms may be imperceptible to all but professionals, except for dyskinesia (involuntary choreic movements). During OFF periods, Parkinson's symptoms either re-emerge or worsen. PD patients can experience the following symptoms: tremor, stiffness, bradykinesia, postural alteration, lack of muscular coordination, difficulty handwriting and freezing of gait (FoG). Although the disease experience is variable; patients in a moderate or severe stage of the disease usually cycle between ON and OFF periods from three to four times a day. With the help of their neurologists, patients need to learn to schedule their medication and their activities around their ON/OFF fluctuations.

Time-in-OFF is the main parameter used to evaluate the effectiveness of a pharmacological intervention and to compare the action of different active principles in clinical trials. Thus, collecting precise information on the temporal course of OFF periods (onset and duration) is essential for adjusting an optimal therapy schedule aimed at preventing motor fluctuations in PD patients. In this sense, the most currently employed method to collect such information is asking PD patients to keep an ON/OFF diary. However, this method has several limitations, including recall bias and reduced compliance, which makes it unsuitable for medium and long term monitoring in clinical practice [2]. Therefore, it would be of great help a portable electronic sensor which could record patient's motor fluctuations reliably and automatically. Furthermore, the use of such devices would enable doctors to personalize medication intakes and, therefore, improve the response to treatment. Additionally, the knowledge of the ON/OFF cycles evolution can be a good indicator of the disease advance for neurologists.

Inertial sensors based on Micro-Electro-Mechanical Systems (MEMS) have been widely used to analyze PD symptoms and infer motor states. A remarkable recent study was presented in 2010 by Zwartjes et al. [3] , in which motor activities and severity related to tremor, bradykinesia and hypokinesia were analyzed in six PD patients, who wore four sensors at wrist, thigh, foot and sternum. Results showed that the method's output was correlated with Unified Parkinson's Disease Rating Scale (UPDRS) values. Another important study was presented by Salarian et al. [4], who detected and quantified tremor and bradykinesia in 20 PD patients by using two triaxial gyroscopes located on each of the forearms. Results also revealed a high correlation with related UPDRS values. Some works have been presented in the lasts years from the European project PERFORM [5][6]. In this works, accuracies up to 86% are achieved with datasets generated among twenty patients performing a set of daily live activities, classifying the presence and severity of bradykinesia, tremor and dyskinesia. However, the system is composed by a set of 5 wearable sensors and a central store unit being, somehow, a cumbersome system for the continuous monitor of daily life activities. So far, the experiments with motor states detectors based on inertial sensors were carried out while the patient wore a set of sensors (generally, several sensors located on different parts of the body) and most of the cases in controlled settings (laboratory), where the patients were asked to perform specific activities

[7][8][9][10]. Finally, in some of these experiments, OFF periods were artificially induced by prolonged withdrawal of the patients' habitual medication, these periods are deeper than the natural OFF periods. Therefore, the algorithms validated under controlled conditions to detect natural OFF periods in daily life activities cannot be assured.

This work presents a new method for the detection of ON and OFF motor states by using a single waist-worn accelerometer and its preliminary results in a population of 7 PD patients. The device and method were tested during their daily life and without any withdrawal of medication. This study is part of the Monitoring the Mobility of PD Patients for Therapeutic Purposes 2 (MoMoPa2) project, in which will evaluate this method in 15 patients. The relevant results presented in this paper from the first 7 patients show that an accurate monitoring of motor fluctuations can be obtained from a single waist-worn device.

2 Related Work

The research conducted so far on the ON/OFF motor states detection by means of inertial sensors is mainly based on the patient symptoms characterization. This line is followed by the first work conducted in 2004 by scientists of the Medical Center at Leiden University [10]. This study focused on determining the motor state of 50 PD patients through the characterization of hypokinesia, bradykinesia and tremor, which are closely related to the OFF motor state. To this end, two triaxial accelerometers located on the wrists were employed and these symptoms were characterized through three signal features: time the patient spent immobile, mean acceleration and percentage of time with tremor respectively, each calculated in intervals of 30 minutes. A linear classifier was used to determine the patient's motor state, so that a threshold is applied to each feature and, thus, both motor states can be separated. Results demonstrate a relatively good performance since they achieved a specificity value (accuracy rate in ON detection) equal to 0.7 and a sensitivity value (accuracy rate in OFF detection) close to 0.7, too.

Researchers at the École Polytechnique Fédérale de Lausanne evaluated the use 7 gyroscopes and 2 accelerometers located on the forearms, shins and trunk detecting motor states [11]. They evaluated the use of 15 parameters and subsets of them representing the presence or absence of tremor, bradykinesia, postural transitions, body posture and gait parameters to determine the motor state of 13 patients, using logistic regression. They used 10-minute windows in tests lasting from 3 to 6 hours. The results showed that, for a subset of 9 features, a specificity of 0.9 and sensitivity of 0.76 were achieved.

Cancela et al. present a motor symptom monitoring and management system developed in a project called PERFORM [6]. In this work a set of classification algorithms were evaluated obtaining accuracy of 86%. The database was generated with twenty patients performing a set of activities of daily live. Pastorino et al. used a version of Cancela's algorithm with a database from twenty-four patients performing unconstrained and unscripted activities at their homes for a week [5]. The result present an accuracy of 74.4±14.9%. Nevertheless, in both cases, the system is

composed by a set of 5 wearable sensors and a central store unit making the system unusable as a continuous monitor for the daily life activities.

On the other hand, drug-induced dyskinesia, which is not a PD symptom but a side effect of the dopaminergic treatment, has been widely studied in order to infer patients' ON motor state. Drug-induced dyskinesia refers to the involuntary adventitious movements that commonly occur when antiparkinsonian effects of treatment are maximal, hence the term "peak dose dyskinesia". In terms of motor fluctuations the so called peak dose dyskinesia is a conclusive motor disorder related with the ON states. In this sense, many works have analyzed their automatic detection based on MEMS sensors. Recently, Keijsers et al. were able to achieve an accuracy of 96.6% in detecting dyskinesia in thirteen patients with six tri-axial acceleration sensors [12]. The employed algorithm classified fifteen minute segments using a regular neural network. In contrast, Tsipouras et al. were able to achieve similar results but on smaller segments [13]. Their dataset contained inertial sensor data of four patients and six control subjects who performed a set of predefined activities. In total, two gyroscopes and six accelerometers distributed on the patient's body were used during recording sessions. In this work, five classification methods were evaluated, obtaining accuracies from 53.85% to 93.7%.

A distinct tendency for the evaluation of patient's motor state involves scoring patient response to some exercises similar to those performed in the evaluation of the UPDRS scale from the sensor signal. The UPDRS score provides a value that measures the severity in which PD is affecting a patient at the time of evaluation. This way, motor fluctuations can be monitored because there is a close relationship between the UPDRS value and ON/OFF motor state [14]. For instance, Kinesia device, which was developed by the American company Great Lakes Technology, consists of a triaxial accelerometer and a gyroscope which measures patient's finger movement. This device is capable of providing measures correlated with UPDRS [15]. Another example is the work of researchers from Harvard Medical School, who used 9 accelerometers to estimate the value of UPDRS in different hand, feet and heels exercises [8]. The latter two methods have the disadvantage of requiring the patient to perform specific exercises other than those usually performing during daily life. This way, the capabilities of capturing the rapid fluctuations of the disease are limited. Furthermore, these experiments force patients to stop their normal activity.

As a conclusion, currently there is not any available minimally invasive system able to monitor ON and OFF motor states during patients' daily life. Moreover, as it was mentioned in the introduction, in most of previous works, OFF periods were artificially induced by prolonged withdrawal of the patients' habitual medication, which commonly provoke deeper OFF periods than the natural ones and could result in a non-real conditions evaluations. In the next section, a new method for motor states detection in PD patients based on a single waist-worn sensor is presented, which is evaluated in 7 PD patients without medication withdrawal.

3 ON/OFF Detection Based on a Single Sensor on the Waist

Determining the ON/OFF state of a patient is a highly complex problem due to the involvement of, as a relevant factor, subjectivity from patient's side. In this work, the proposed ON/OFF detection, similarly to other state-of-the-art methods, employs the output of specific online algorithms that evaluate different movement disorders associated to the state of a PD patient. Their outputs are combined in a hierarchical evaluation in order to establish the motor state of the patient. In this proposal, ON/OFF states are detected based on dyskinesia and bradykinesia detection, according to the description given below.

Generally, it is assumed that motor state fluctuations are associated with the fluctuations of dopamine levels in the brain, which is, at the same time, strongly related to PD motor disorders. The motor symptom that has been shown to maximally correlate with the dopaminergic deficit is bradykinesia; more specifically, the less dopamine level produces more bradykinesia, which is associated with the OFF state. Another motor disorder that is highly related to the level of dopamine is the choreic peak-dose dyskinesia. This disorder appears with high levels of dopamine in the brain and is associated with ON states [1]. In order to determine the presence of these motor disorders that are characteristic of ON and OFF states in an unobtrusive way, it is taken into account that motor activities in daily life are a sequence of movements that have different degrees of automation and, furthermore, these automatic movements are slower in PD patients due to an increase of startup time of the movement and a slow implementation of the movement [16]. At the same time, the most characteristic automatic movement in humans is gait. In consequence, in this work, it is considered that gait's high degree of automation makes gait analysis the optimal tool to analyze bradykinesia, understood as the absence, slowed and/or progressive reduction of movement. In contrast, in current clinical practice, the automatic movements are evaluated, for simplicity, with simple repetitive and voluntary movements. However, our aim is to monitor patients in daily life without requiring any pre-established movement so, in this sense, evaluating gait in daily life allows us to develop a real-time algorithm minimizing the impact in the patients activities. On the other hand, dyskinesia is a hyperkinetic disorder and becomes more evident and clear when the patient is at rest. Due to this reason, dyskinesia is evaluated when patients do not walk. As a result, both algorithms provide information in complementary situations.

Next sections present each algorithm that comprises the overall motor state detection. It should be noted that, given the minimum-invasive nature of the proposed system consisting in a single device that should execute all algorithms in real-time, the development of the algorithms had, as one of the main objectives, economize the computational and memory resources of the hardware system. Consequently, the resulting device is an online monitor with very low consumption.

3.1 Dyskinesia Detection

The first algorithm, in which ON/OFF monitoring relies on, detects dopaminergic-induced dyskinesia. This algorithm was developed after observing that, according to literature, frequency is the most relevant feature of inertial signals provided by inertial

sensors that enables the characterization of dyskinesia. In this sense, it is observed that power spectral density in the dyskinetic frequency band, approximately between 0 and 4 Hz, is increased during dyskinesia. However, other activities, such as gait or using stairs, may also increase this frequency band, as described in [17]. According to these observations, dyskinesia assessment is based on splitting the spectra into three different bands: dyskinetic band, covering the (0.68, 4] Hz range; non-dyskinetic band, considered to be [8, 20] Hz; and Posture Transition (PT) band, considered to be (0, 0.68] [18]. Since 20 Hz is the maximum frequency analyzed, sampling frequency is set to 40 Hz.

This algorithm enables us to know whether in a given time window of 3.2 s the patient suffered dyskinesia or walked based on the defined spectra band. If the power spectrum in the non-dyskinetic band exceeds a pre-defined threshold, it is concluded that the patient is walking and, in consequence, the presence of dyskinesia cannot be determined since gait movement hides it. Otherwise, the presence of dyskinesia is inferred if the dyskinetic band exceeds a second threshold. Both thresholds were determined on a previous study [19].

However, dyskinesia is a symptom which may last several minutes. The most suitable window to examine the appearance of dyskinesia is, then, of several windows length. Thus, it is proposed to aggregate the output obtained in many windows of 3.2 s length overlapped at 50% along a period of 60 seconds. Finally, the output of the algorithm for a one-minute period covering a certain minute i, which is noted as d_i, is then classified as dyskinetic (d_i =1) in case the majority of the windows along this period were dyskinetic as well. In case most of them were not dyskinetic the whole period is classified accordingly (d_i =-1). If none of the windows within the one-minute period were analyzed, i.e. the patient walked during the totality of the minute, the output is undetermined (d_i =NaN). The algorithm used to detect the dyskinesia status has been published in [19].

3.2 Bradykinesia Detection

The method followed by the online algorithm to determine the presence of bradykinesia is based upon analyzing patients' motion fluency while walking. This way, bradykinesia analysis relies on the characterization of gait cycles, i.e. strides. A prerequisite is that patient should be walking; thus, a gait detector should enable gait cycles analysis. The latter analysis should perform, on the one hand, an identification of gait cycles from the accelerometer signal, i.e. stride detection, and, on the other hand, should characterize gait cycles by some features that correlates bradykinesia. Complete bradykinesia analysis consist in a four-step characterization method.

The first step is based on a Support Vector Machine (SVM) for detecting walking periods in 3.2 second windows, as in the dyskinesia case. The output was used to classify every window as "walking" or "not walking". The second phase was focused on detecting patients' strides is based on Zijlstra et al. [20] work adapting the location of the sensor. The third step is the characterization, the strides detected were analyzed and the resulting measurement consists of the representation of a stride, which was proportional to patient's motion fluency. Finally, the values obtained in the third phase are averaged within a period of 60 seconds. The averaged value was compared with two different thresholds (high and low), previously set in a particularized session

with every patient, in order to determine the patient's motor status at that moment. If the minute-averaged measurement was higher than the high threshold, it was considered that the patient did not present bradykinesia. If the averaged measurement was between both high and low thresholds, then it was considered that the patient presented an intermediate state.

Thus, the output of the bradykinesia algorithm in a given one-minute period i, noted as b_i, is b_i=NaN in case the patient did not walk in the corresponding minute, b_i=1 in case bradykinesia was detected (averaged value higher than the higher threshold), b_i=0 in case of an intermediate state (averaged value between thresholds) and b_i= –1 in case of no bradykinesia (averaged value below lower threshold). The threshold was adapted individually by using data available from the patient acquired in a particularization day provided in the clinical study. The algorithm used to detect bradykinesia status was published in [17] and [21].

3.3 ON/OFF Motor States Detection

As stated in the introduction, during OFF states, various movement disorders that alter patients' gait may appear: rigidity, bradykinesia, FoG, festination and postural instability. During ON states, symptoms are alleviated and patients move more smoothly. Moreover, involuntary movements from dopaminergic-induced dyskinesia may be suffered during ON periods. In this study, ON and OFF motor states detection is focused on both aspects: gait analysis and dyskinesia. To this end, a hierarchical classifier is employed to combine the information from both algorithms in order to determine current motor state of a patient.

The proposed motor states classifier is, initially, automatically particularized to each patient, based on basic information of the individual disease symptomatology. This way, for those patients who do not present peak-dose dyskinesia, their corresponding motor state detection would not consider the output of the dyskinesia algorithm.

Motor states classifier employs the output provided every minute by both bradykinesia and dyskinesia algorithms. Given the nature of motor fluctuations, which can happen few times a day, the output of several minutes must be considered. To this end, two different trees that cover two different time windows, 10 and 30 minutes, have been evaluated. More concretely, the motor states classifier tree recalculates bradykinesia and dyskinesia algorithms, noted as b and d, respectively, according to the following equations:

$$b = \begin{cases} NaN & b_i = NaN \ \ \forall \ i = 0..x_l \\ -1 & \|b_i = -1\| > \|b_i = 0\| \ \wedge \|b_i = -1\| > x_b, \ \forall \ i \ = 0..x_l \\ 1 & (\|b_i = -1\| + \|b_i = 0\|) < \|b_i = 1\| \ \wedge \|b_i = 1\| > 2, \ \forall \ i = 0..x_l \\ 0 & otherwise \end{cases} \quad (1)$$

$$d = \begin{cases} NaN & \|d_i = NaN\| > x_{dNaN}, \quad \forall \ i = 0..x_l \\ 1 & \|d_i = 1\| \geq x_d, \quad \forall \ i \ = 0..x_l \\ 0 & otherwise \end{cases} \quad (2)$$

where $\wedge$ is the logical *AND* operator and $\|\cdot\|$ counts the number of elements satisfying the within condition. The interval of time in which the algorithm determines

whether dyskinesia and bradykinesia are present or not is represented by x_l, which takes values 9 and 29 to establish a window length of 10 and 30 minutes, respectively. The values for the x_b, x_{dNan} and x_d have been empirically set and are 2, 7 and *3* correspondingly for a window length of 10 minutes and 4, 21 and 9, respectively, for a window length of 30 minutes.

The meaning of b is the same that b_i: value 1 corresponds to bradykinesia, 0 to intermediate state and -1 to non-bradykinetic period. In equation (1), b cannot be evaluated if all b_i outputs are *NaN*, meaning that there were no strides within the period. Similarly, the meaning of d is the same that d_i: value 1 corresponds to dyskinesia, 0 to non-dyskinetic period. In equation (2), d can only be evaluated if the major percentage of the d_i outputs are evaluated, otherwise, it is assigned to *NaN*.

The motor states classifier, that determines the ON/OFF state, is evaluated every time window based in the output comprised within the period. If the patient does not present dyskinesia in the clinical record, the output will only be the bradykinesia regrouped depending on the different time windows. If the patient presents peak dose dyskinesias, then, the general algorithm presented below is applied:

$$ON/OFF \begin{cases} NaN\ (not\ evaluated) & if (b = NaN \ \wedge \ d = NaN\ ) \ \vee (b = 1 \ \wedge \ d = 1) \\ -1\ \text{(ON state)} & if\ b = -1 \ \vee \ d = 1 \\ 0\ \ \text{(Intermediate state)} & if\ b = 0 \ \wedge \ d \neq 1 \\ 1\ \ \text{(OFF state)} & if\ b = 1 \ \wedge \ d \neq 1 \end{cases}$$

where $\vee$ is the logical *OR* operation.

The motor state inferred is ON when the bradykinesia is not present (b=-1) or when dyskinesia output d is 1 (present). This is due to, on the one hand, a lack of symptomatic gait in ON states and, on the other hand, if the patient has not walked, because of dyskinesia detection. The OFF state is determined as the detection of bradykinesia and the absence of dyskinesia. Finally, the intermediate state is defined by an intermediate detection of the bradykinesia algorithm and lack of dyskinesia. In the final output of the ON-OFF decision tree, a temporary filter that fills empty periods, i.e. *NaN*-evaluated, of 10 or 30 minutes is applied, although not evaluated. If a blank period is found between two periods whose states were determined and equal (either ON or OFF), it will be filled with the same state. Instead, if the adjacent periods to an empty period were determined and not equals, an intermediate state is then inserted.

4 Experiments

4.1 Participants, Materials and Experimental Protocol

In this experiment, a total of 7 PD patients with idiopathic PD according to UK PD Society Brain Bank criteria [22] have participated. Ages ranged from 49 to 82 years and patients were in a mild or moderate stage of the disease. In order to be sure that a correct diary was filled, only those patients capable of recognizing their motor state were recruited. The experimental protocol was approved by the local Ethics Review Committee.

The experiment had two stages divided in two different days. The first day consist of a particularization of the parameters system. To this end, patients performed various activities while an inertial sensor located at their waist registered acceleration and a video camera registered the activities in order to generate the gold-standard for the particularization process. This day was divided into two sessions marked by a clinically defined OFF/ON states. Prior to any recording, the motor state was verified and the motor section of the UPDRS was evaluated. Afterwards, the recording device was switched on and recording started. Participants performed a series of tests composed of different walking activities. A similar procedure was repeated for the ON recording session. Again, the participant was recorded and the motor section of the UPDRS is also evaluated at the beginning of this session.

Data collected on the first day are employed to particularize the threshold used in the bradykinesia algorithm. These data are labeled either with the tag of bradykinesia present or not present. The threshold is, then, obtained as the value that separates the data with a maximum distance of both sets. The process is explained in detail in [23].

The second day, a sensor was given to each user after describing how it works. Then, the patient was left alone with the sensor recording and performing the corresponding usual daily life activities. Along the second day, patients were asked to fill in a diary of their own motor state every 30 minutes. Finally, every two hours a researcher called the patient and asked for the current motor state, generating an alternative diary in order to complement that filled in by the patient.

A small and wearable device developed at CETpD was used to register the accelerometer measures during the first day and to register the output of the online algorithms during the second day. The device included a tri-axial accelerometer (±6g range), a dsPIC33F microcontroller, a µSD Card and a Li-ion battery [24].

4.2 Ground-Truth Based on Motor States Diary

The main problem associated to PD patient diaries is twofold. First, patients might not be able to recognize their own motor state and symptoms. Second, non-motor symptoms of PD may introduce false data in the diaries. Moreover, patients often forget to write some motor state changes and may put them in the wrong time slot. Overall, patient diaries are commonly not reliable, for which a call every two hours was included in order to check the current state of patient.

In order to minimize the impact of these issues, the evaluation of the motor state decision tree is performed taking into account two different timing strategies with respect to patient diaries. On the one hand, we have worked with a so-called "strict" strategy. In this strategy, only consecutive annotations of equal motor state are analyzed. This strategy ensures, with high probability, that the patient has not changed the motor state during this period, but also rejects many annotations. On the other hand, a "slack" diary has been implemented, which has been given a temporary validity for 15 minutes ahead and behind to each annotation. Finally, in each strategy, only those annotations that were validated in the phone call are considered.

5 Results

A total of 40.33 hours of inertial sensor signals were recorded, which corresponded to the motion records of the 7 subjects who participated in the experiment. As an example, at the top of Figure 1 is presented the "strict" version of the patient diary of Patient 1 versus the sensor output validated in the valid labels of the diary (only between two equal motor states) in a 30 minutes window. At the bottom of Figure 1 is presented the patient diary of Patient 6 versus the sensor output in a 10 minutes window.

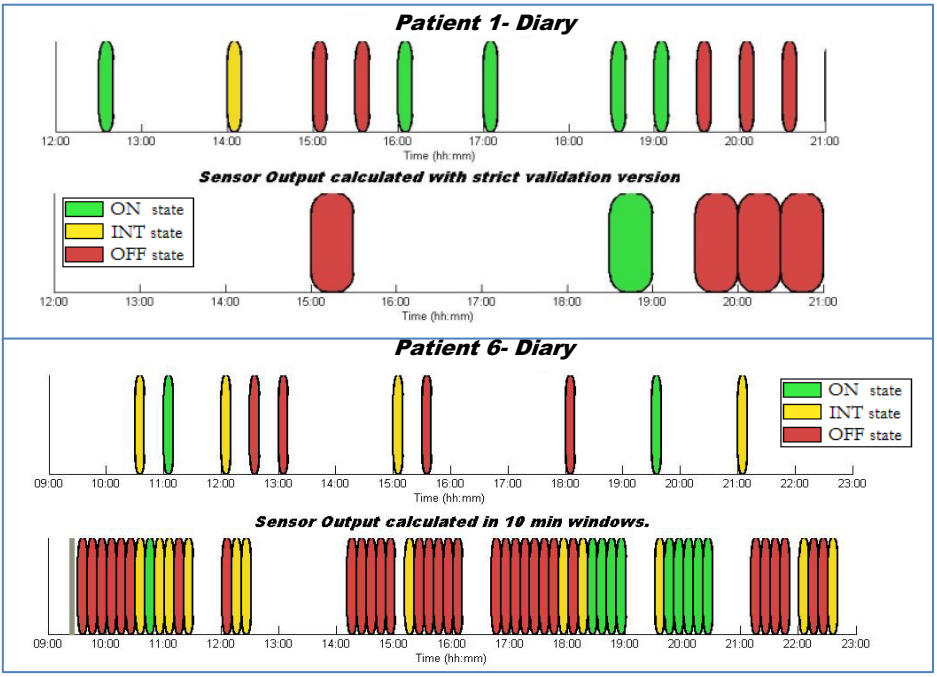


Fig. 1. Examples of patient diary versus the sensor output

Table 1 shows the sensitivity, specificity, number of detections produced by algorithm (Total outputs), the number of these outputs that could be validated in the diary (Outputs used), the number of ON, OFF and intermediate states detected (Outputs by state), the number of valid labels in the diary (i.e. annotations written during day) (Total labels), the number of valid labels used in the validation, (i.e. annotations used in the validation) (Labels used) and the number of ON, OFF and intermediate states in the diary (Labels by state).

Table 1. Summary of results organized by patient (Pat), strategy and window time

Window time	Timing strate-gy	Pat.	Spec.	Sens.	Outputs		Outputs by state			Labels		Labels by state		
					Used	Total	OFF	ON	INT	Used	Total	OFF	ON	INT
30'	Strict strategy	**1**	1.00	1.00	5	5	4	1	0	5	6	5	1	0
		2	1.00	1.00	2	2	1	1	0	2	3	1	2	0
		3	1.00	1.00	3	10	1	2	7	3	10	3	7	0
		4	1.00	1.00	4	6	1	3	2	4	9	4	5	0
		5	NaN	NaN	0	0	0	0	0	0	1	1	0	0
		6	1.00	NaN	6	14	0	6	8	6	16	1	15	0
		7	1.00	NaN	1	2	0	1	1	1	2	0	2	0
	Slack strategy	**1**	1.00	1.00	6	8	4	2	2	6	13	7	5	1
		2	1.00	NaN	1	1	0	1	0	1	7	2	4	1
		3	1.00	1.00	5	13	1	4	8	5	17	6	11	0
		4	1.00	1.00	5	6	1	4	1	5	10	5	5	0
		5	1.00	1.00	3	5	1	2	2	3	5	2	1	2
		6	1.00	NaN	6	15	0	6	9	6	20	2	17	1
		7	1.00	NaN	1	4	0	1	3	1	7	1	5	1
10'	Strict strategy	**1**	1.00	1.00	8	49	10	24	15	4	6	5	1	0
		2	1.00	1.00	3	7	2	1	4	1	3	1	2	0
		3	1.00	1.00	19	36	1	12	23	9	10	3	7	0
		4	1.00	1.00	6	15	3	6	6	3	9	4	5	0
		5	NaN	NaN	0	55	29	10	16	0	1	1	0	0
		6	1.00	NaN	13	41	0	15	26	8	16	1	15	0
		7	1.00	NaN	1	39	1	26	12	1	2	0	2	0
	Slack strategy	**1**	1.00	0.90	16	49	10	24	15	8	13	7	5	1
		2	1.00	1.00	3	7	2	1	4	1	7	2	4	1
		3	1.00	1.00	24	36	1	12	23	11	17	6	11	0
		4	0.83	1.00	7	15	3	6	6	4	10	5	5	0
		5	NaN	1.00	3	55	29	10	16	2	5	2	1	2
		6	1.00	NaN	14	41	0	15	26	8	20	2	17	1
		7	1.00	NaN	4	39	1	26	12	3	7	1	5	1

Table 1 presents the results obtained by the algorithm based on 7 PD patients at their homes without supervision and performing their usual daily life activities. The average sensitivity and specificity for both time windows (10 min and 30 min) are 99.9% and 99.9% respectively. According to the results shown, it is observed that 10-minutes window provides slightly worse results than the 30-minutes window. However, the main difference between the diary strategies is, clearly, the evaluated time in the calculation of sensitivity and specificity due to the conservative strategy of the strict diary. The average time evaluated by patient in the "strict" strategy is 80 minutes, while for the "slack" strategy is 108 minutes.

So far, other works attempted to detect the patient's motor status. However, to our knowledge, they were carried out while the patient wore a set of sensors (generally, several sensors located on different parts of the body) and most of the cases in controlled settings (laboratory), where the patients were asked to perform specific activities [5][6][7][8][9]. Hoff et al. [10] also reported validation results during daily life activities. However, their specificity and sensitivity values were lower than ours.

A limitation of our motor status (ON/OFF) detection algorithm is related with the fact that one of the main parts of its analysis is based on patients' walking. This may lead to the occurrence of long periods without motor information while the patient is in OFF state. This is especially critical because patients in OFF tend to stay still. However, PD patients in moderate and advance stages of the disease, use to walk more than 40 times per day [25][26] and our system produce enough frequent detections. This system could provide very useful information for a better clinical monitoring. Currently, no detection system at all is available and the neurologists having to depend on patients' or caretakers' reports. Definitely, a system such as that presented in this work would be seen as a big step forward.

6 Conclusions

In the present study, an algorithm for ON/OFF detection based on bradykinetic gait and choreic dyskinesia detection has been presented and validated in a short number of patients with idiopathic Parkinson's disease, on the basis of inertial signals produced by a sensor attached to the patient's waist. The magnificent specificity and sensitivity values found lead us to be very optimistic about the performance of this algorithm despite the low number of patients that has been validated for now. These relevant results are going to be reinforced, at the end of the MoMoPa 2 study, with the validation on 15 PD patients.

Main advances that automatic detection of PD motor status can provide consist of obtaining accurate information for physicians in order to adjust the medication intake and, moreover, the possibility of modifying drug infusion rates in apomorphine and duodopa pumps. The possibility of objectively mapping the motor status of PD patients may additionally promote the use of such detectors in clinical trials to reliably determine subjects' responses to experimental medication.

In conclusion, the use of portable devices for monitoring motor fluctuations by PD patients in their habitual environment is a great advance in the clinical practice. The use of such devices would open the way to enhance control of pharmacological therapy and to interact with other electronic devices such as drug infusion pumps.

Acknowledgements. This work has been performed within the framework of the MoMoPa2 Project (Monitoring the mobility of Parkinson's patients for therapeutic purposes 2 – PI12/03028) funded by the Instituto de Salud Carlos III – Ministerio de Economía y Competividad and the European Regional Development Fund (ERDF).

References

1. Jankovic, J.: Parkinson's disease: clinical features and diagnosis. J. Neurol. Neurosurg. Psychiatry **79**(4), 368–376 (2008)
2. Papapetropoulos, S.S.: Patient diaries as a clinical endpoint in Parkinson's disease clinical trials. CNS Neurosci. Ther. **18**(5), 380–387 (2012)
3. Zwartjes, D., Heida, T., van Vugt, J., Geelen, J., Veltink, P.: Ambulatory Monitoring of Activities and Motor Symptoms in Parkinson's Disease. IEEE Trans. Biomed. Eng. **57**(11), 2778–2786 (2010)
4. Salarian, A., Russmann, H., Wider, C., Burkhard, P.R., Vingerhoets, F.J.G., Aminian, K.: Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system. IEEE Trans. Biomed. Eng. **54**(2), 313–322 (2007)
5. Pastorino, M., Cancela, J., Arredondo, M.T., Pansera, M., Pastor-Sanz, L., Villagra, F., Pastor, M.A., Martin, J.A.: Assessment of bradykinesia in Parkinson's disease patients through a multi-parametric system. In: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, pp. 1810–1813 (2011)
6. Cancela, J., Pansera, M., Arredondo, M.T., Estrada, J.J., Pastorino, M., Pastor-Sanz, L., Villalar, J.L.: A comprehensive motor symptom monitoring and management system: The bradykinesia case. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1008–1011 (2010)
7. Keijsers, N.L.W., Horstink, M.W.I.M., Gielen, S.C.A.M.: Ambulatory motor assessment in Parkinson's disease. Mov. Disord. **21**(1), 34–44 (2006)
8. Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J., Standaert, D., Akay, M., Dy, J., Welsh, M., Bonato, P., Member, S.: Monitoring Motor Fluctuations in Patients With Parkinson's Disease Using Wearable Sensors. IEEE Trans. Inf. Technol. Biomed. **13**(6), 864–873 (2009)
9. Salarian, A., Russmann, H., Vingerhoets, F.J.G., Dehollain, C., Blanc, Y., Burkhard, P.R., Aminian, K.: Gait assessment in Parkinson's disease: toward an ambulatory system for long-term monitoring. IEEE Trans. Biomed. Eng. **51**(8), 1434–1443 (2004)
10. Hoff, J.I., van der Meer, V., van Hilten, J.J.: Accuracy of Objective Ambulatory Accelerometry in Detecting Motor Complications in Patients With Parkinson Disease. Clin. Neuropharmacol., **27**(2) (2004)
11. Salarian, A.: Ambulatory monitoring of motor functions in patients with parkinson´s disease using kinematic sensors. École polytechnique federale de Lausanne (2006)
12. Keijsers, N.L.W., Horstink, M.W.I.M., Gielen, S.C.A.M.: Automatic assessment of levodopa-induced dyskinesias in daily life by neural networks. Mov. Disord. **18**(1), 70–80 (2003)
13. Tsipouras, M.G., Tzallas, A.T., Rigas, G., Bougia, P., Fotiadis, D.I., Konitsiotis, S.: Automated Levodopa-induced dyskinesia assessment. In: 2010 Annual International Conference of the IEEE on the Engineering in Medicine and Biology Society (EMBC), pp. 2411–2414 (2010)
14. Stebbins, G.T., Goetz, C.G., Lang, A.E., Cubo, E.: Factor analysis of the motor section of the unified Parkinson's disease rating scale during the off-state. Mov. Disord. **14**(4), 585–589 (1999)
15. Mera, T.O., Heldman, D.A., Espay, A.J., Payne, M., Giuffrida, J.P.: Feasibility of home-based automated Parkinson's disease motor assessment. J. Neurosci. Methods **203**(1), 152–156 (2012)
16. Bloxham, C.C., Mindel, T.C., Frith, C.D.: Initiation and execution of predictable and unpredictable movements in parkinson's disease. Brain **107**(2), 371–384 (1984)

17. Samà, A., Perez-Lopez, C., Romagosa, J., Rodriguez-Martin, D., Català, A., Cabestany, J., Perez-Martínez, D.A., Rodríguez-Molinero, A.: Dyskinesia and motor state detection in Parkinson's Disease patients with a single movement sensor. In: 34th Annual International Conference of the IEEE on the Engineering in Medicine and Biology Society, EMBS 2008, pp. 1194–1197 (2012)
18. Rodriguez-Martin, D., Samà, A., Perez-Lopez, C., Català, A., Cabestany, J., Rodriguez-Molinero, A.: SVM-based posture identification with a single waist-located triaxial accelerometer. Expert Syst. Appl. **40**(18), 7203–7211 (2013)
19. Sama, A., Perez-Lopez, C., Romagosa, J., Rodriguez-Martin, D., Catala, A., Cabestany, J., Perez-Martinez, D.A., Rodriguez-Molinero, A.: Dyskinesia and motor state detection in Parkinson's Disease patients with a single movement sensor. In: 2012 Annual International Conference of the IEEE on the Engineering in Medicine and Biology Society (EMBC), pp. 1194–1197 (2012)
20. Zijlstra, W., Hof, A.L.: Assessment of spatio-temporal gait parameters from trunk accelerations during human walking. Gait Posture **18**(2), 1–10 (2003)
21. Rodr'iguez-Molinero, A., Samà, A., Pérez-Mart'inez, A.D., Pérez López, C., Romagosa, J., Bayés, À., Sanz, P., Calopa, M., Gálvez-Barrón, C., de Mingo, E., Rodr'iguez Mart'in, D., Gonzalo, N., Formiga, F., Cabestany, J., Catalá, A.: Validation of a Portable Device for Mapping Motor and Gait Disturbances in Parkinson's Disease. JMIR mHealth uHealth **3**(1), e9 (2015)
22. Vapnik, V.N.: The Nature of Statistical Learning Theory, Second. Springer-Verlag, New York (1995)
23. C.A. Rodríguez-Molinero, A., Samà, A., Pérez-Martínez, D.A., Pérez-López, C., Romagosa, J., Bayés, A., Sanz, P., Calopa, M., Ruiz, J., Gálvez, C., de Mingo, E., Rodríguez-Martín, D., Gonzalo, N., Formiga, F., Cabestany, J.: Validation of a Portable Device for Mapping Motor and Gait Disturbances in Parkinson's Disease. JMIR MHEALTH UHEALTH - Press
24. Rodríguez-Martín, D., Pérez-López, C., Samà, A., Cabestany, J., Català, A.: A Wearable Inertial Measurement Unit for Long-Term Monitoring in the Dependency Care Area. Sensors **13**(10), 14079–14104 (2013)
25. Cavanaugh, J.T., Ellis, T.D., Earhart, G.M., Ford, M.P., Foreman, K.B., Dibble, L.E.: Capturing Ambulatory Activity Decline in Parkinson Disease. J. Neurol. Phys. Ther. JNPT **36**(2), 51 (2012)
26. Rochester, L., Chastin, S.F.M., Lord, S., Baker, K., Burn, D.J.: Understanding the impact of deep brain stimulation on ambulatory activity in advanced Parkinson's disease. J. Neurol. **259**(6), 1081–1086 (2012)

Convolutional Neural Networks for Detecting and Mapping Crowds in First Person Vision Applications

Juan Sebastian Olier[1,2(✉)], Carlo Regazzoni[1], Lucio Marcenaro[1], and Matthias Rauterberg[2]

[1] Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN), University of Genoa, Genoa, Italy
sebastian.olier@ginevra.dibe.unige.it
[2] Department of Industrial Design, Eindhoven University of Technology, Eindhoven, Netherlands

Abstract. There has been an increasing interest on the analysis of First Person Videos in the last few years due to the spread of low-cost wearable devices. Nevertheless, the understanding of the environment surrounding the wearer is a difficult task with many elements involved. In this work, a method for detecting and mapping the presence of people and crowds around the wearer is presented. Features extracted at the crowd level are used for building a robust representation that can handle the variations and occlusion of people's visual characteristics inside a crowd. To this aim, convolutional neural networks have been exploited. Results demonstrate that this approach achieves a high accuracy on the recognition of crowds, as well as the possibility of a general interpretation of the context trough the classification of characteristics of the segmented background.

Keywords: Convolutional neural networks · Crowds detection · First-person vision · Egocentric videos

1 Introduction

The increasing development of wearable devices, and in particular of wearable cameras at a low price, gives rise to unexplored scenarios where many new applications can be developed, along with which many new challenges are posed. In particular, first person vision (FPV) greatly enhances the possibilities of understanding the surroundings of the wearer, but equally implies many problems to be faced in image processing.

When it comes to the interaction with and of people around the wearer of an FPV device (e.g. smart glasses), the perspective from which the video is taken offers a different understanding of the behaviors, since wearers can move and interacts more naturally, and thus produce data from this more fluent and versatile point of view. Therefore, such capability can be used to develop a context

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 475–485, 2015.
DOI: 10.1007/978-3-319-19258-1_39

awareness and understanding processing tool, capable of extracting meaningful analytics with respect to changing environments (in particular crowds) and their dynamic evolution. Such tools could be applied in fields like group and social dynamics understanding, affective computing, surveillance, or assistive technologies. Particularly, in crowded environments this would allow innovative ways of understanding activities in specific areas of interest, as a source of new measures for crowd monitoring and as complement to standard surveillance cameras.

In general, any application based on the ideas mentioned implies the segmentation of crowds and background in the FPV video. Thus, in this paper, a method for the detection of crowds from FPV data is proposed. The estimation of people's positions is addressed as an image classification task with video features extracted directly at the crowd level. The aim of this approach is to provide a processing method to be used for modeling interactions inside the crowds, as well as between them and the wearer of the FPV device.

However, the detection of crowds from this kind of videos poses new ambitious challenges due to the high amount of ego-motion, as well as to the variations in the background and environment that might happen in short periods [5]. Moreover, the irregular features of crowds represent yet another challenge, since many possible occlusions can occur and no particular part of the body, nor any specific visual characteristic, can be expected to be prominent or always present when estimating the presence of a person or a multitude.

Thus, a requirement of the method to be used is that it should be capable of extracting features directly from sample data, and to robustly handle the kind of variations described for the problem at hand. These requirements make Convolutional Neural Networks (CNN) an interesting option, for their capability of learning relevant features from data, and the outstanding performance that has been achieved with them in recent years for image classification tasks as shown in [13] and [6].

Furthermore, the method presented here represents a contribution in itself as, to the best of our knowledge, the problem of mapping crowds around a wearer, from a first person vision perspective, has not been yet addressed.

The remainder of the article is structured as follows; section 2 briefly reviews the state of the art on FPV research. In section 3 the statement of the problem, the architecture and training data are described. In section 4 the results found are described and finally in section 5 conclusions and future work are discussed.

2 Previous Work

The research interest in FPV has been increasing in the last few years along with the development of new technologies and the search for applications as described in [1]. One of the main areas investigated so far is activity recognition [9], where the focus is the assessment of interactions between the wearer and the environment, paying particular attention on the manipulation of objects and hands movements as in [3] and [14].

Similarly, the understanding of situations can be related to the interpretation of interactions with people and the way these evolve dynamically, which in turn

can be used to segment video sequences into relevant periods. However, most of the contributions on scene understanding and segmentation of videos are based on similarities on the actions, situations or environments, with interesting results like the ones found in [8] [12] and [11]. Non has directly focused on the presence and interactions of people as the basic segmentation criteria, being a task which may require a tool for extracting and mapping people like the one proposed in the present work.

Nonetheless, in relation to social interactions, some research has been carried out in works like [4], where the gaze direction is used to infer the attention of the people surrounding the wearer in order to recognize communications and social relations. To this aim face detection is used to detect people, what is done because of the particular goals, but limits the problem to particular kinds of interactions and to the presence of faces which generally represents a small subset of all the possible cases in which a person can appear in a scene.

Equally, in pedestrian's detection, a relevant topic deeply explored in the last decades, most of the contributions have focused on fixed cameras, normally above the crowd, and with the aim of analyzing its dynamics. Likewise, relevant works have recently presented important increases in detection rates by using deep architectures, outperforming most of or all previous methods as in [2] and [10], and even from FPV in [15].

Nevertheless, very limited efforts have been devoted towards the estimation of position of people surrounding the wearer in complex environments and from any perspective, being that all of these approaches emphasize on the detection of persons, searching for the complete shape of the bodies and not directly on the crowds and variable visible parts. That implies a training data focused on people and specific bodily characteristics, parts or poses. On the contrary, the present work focuses on the detection of people and crowds in general without assumptions on particular visible characteristics.

3 Approach

This section describes the general approach through a statement of the general problem, then the selection of the specific method to be used, the data used to train the system, and finally the architecture used.

3.1 General Description of the Problem

Scene understanding from FPV videos is the main goal, with a particular focus on describing the environment and the states of people around a wearer as the main actors in crowded environments. The aim is to provide context awareness and understanding processing tool, with which it would be possible to extract meaningful analytics of changing crowded environments and their dynamic evolution. Additionally, given that the interpretation of the context is not only dependent on the crowd but is also related to the characteristics of the environment, in the extracted background it is possible to find some information about

situations in which the wearer is immersed, which could be, at least partially, analyzed through the classification of certain features of the background.

This way, the main task is to segment the image into crowd and background. However, due to the complex visual appearance of natural scenes and the abrupt variations of background caused by the ego-motion, the detection of people from first person videos represents new challenges, many of which are not addressable through approaches normally adopted for pedestrian recognition or crowd analysis.

In particular, as the situation is highly variable, the position, orientation and movements of the people, as well as the crowd density is not predictable, and then it is not possible to make an assumption about detecting particular parts of the body, or any specific visual characteristic. For instance, there is no guarantee to find body parts in all situations, simply because big parts of the body can be out of the field of view when the person is close to the camera, and probably occluded by other people or objects when far. Consequently, the approach of the present work focuses on the detection of crowds and the extraction of features directly from them.

Finally, the desired result is expected to contain a spatial representation of the crowd's distribution, which might allow mapping and analyzing the situation surrounding the wearer as immersed in a particular scenario. Thus, the method developed cannot only be focused on the detection of the crowds, but must allow the approximation of their position in a 3D-like way, implying the estimation of a plausible distance from the camera.

3.2 Detecting Crowds

The task of estimating the presence of crowds in a particular part of the image, as well as to obtain features that allow to perform classification, are the main goals to have in mind when selecting a method to face the problem at hand. However, many methods that could be suitable for this task may assume some previous knowledge of the features, the presence of some characteristics in all the images, or in some cases the capability of extracting the background, which as described above is not feasible for FPV videos.

Then, the crowd's classification problem implies features that can be extracted only from images containing people and crowds, and on features related to the environment in the case of the background. This statement has led to the selection of Convolutional Neural Networks (CNN) as the main method to address the proposed goal; mainly due to the outstanding performance that has been shown for the CNNs in object recognition tasks and feature extraction from sample data [13][7][6]. Particularly in [13], CNNs are compared to state of the art methods for image recognition, showing how deep architectures outperforms them in most of the datasets used. Equally, in [7] the detection of pedestrians using an architecture based on deep CNNs, in general outperforms or performs comparably to the state of the art methods in the most challenging pedestrian detection data sets.

Additionally, the separation between the crowd and the background resulting of the segmentation, leaves the opportunity to further analyze the information

in the image in order to infer more characteristics of the situation and the environment. Thus, the possibility of extracting such kinds of information from the background are taken into account when designing the network and the dataset to train it.

Design of the Training Data. In order to train a CNN diverse approaches can be explored, from whole frames containing different amount of crowds, aiming to classify them by the density found in them, to the estimation of crowds inside small patches from the image. Obviously, the first option turns out to be impractical in many cases, and as the main features to learn are the ones specific to the crowds, using small patches of the images containing them becomes the best option. Extracting patches is beneficial as well when building the data set to train the network, for it makes it possible to easily include variability as a single image containing a crowd may be sampled in such a way that many patches with different variations can be created.

Patches from crowds and background are used to train a CNN, nonetheless, the variability of the features contained in the background is much higher than the ones in people and multitudes, and in some cases even similar to the ones in crowds. Thus the number of background samples must be higher than the one of crowds in order to achieve a good performance.

The set of crowd images has been split into three classes based on the distance of the crowd to the camera. Hence, the classes "Crowds", "Mid-distance crowds" and "Far crowds" are created. This approach can reduce the variability in the features, but also allows estimating the position of the crowd in three dimensions.

The definition of these three classes is done by assuming a 4 to 1 ratio as a regular proportion of the height of a person and the width of the torso. Assuming this ratio, and considering the closeness to the people that a wearer can have when immersed in a crowd; the first distance is assumed to include at most the forth part of the height of a person, and therefore the whole width of the torso. Subsequently, the next distance doubles the previous one, meaning half of the height. Then the third, doubling again, includes the whole height. In this way, the distance to the camera follows a linear function of the portion of height included, where each one is the double of the previous, and the basis depends on the angular view of the camera and the size of the patches.

Having a higher number of samples for background also allows to create different classes for it. Such classes can be selected in different ways, but finding the best division is beyond the scope of this paper; the dataset for background has been split into six different classes (Plants, Interior, Sky, Buildings, Exterior and Floor). With such division, it is possible to estimate other kind of information present in natural scenes representing basic cues of the context and the environment, and could be used even for a better estimation of possible activities.

Consequently, a set of images to train a CNN with these classes has been created based on FPV videos dataset presented in [4] and other images from different publicly available sources. The set is balanced in the number of images per class and contains a total of 57.600, out of which a 15% is used for validation.

Fig. 1. Samples of the three classes of crowds used to train the CNN. a) "Crowds", b) "Mid-distance crowds", and c) "Far crowds"

Fig. 2. Samples of the six classes of background used to train the CNN. Plants, Interior, Sky, Buildings, Exterior and Floor

Architecture of the CNN. The network used has 3 convolutional layers and 2 fully connected ones, and inputs used are 50 by 50 pixels RGB images (the mentioned patches). All convolutional layers are followed by Rectified Linear Units ('ReLU'), and the first two are followed by a 2 by 2 max pooling. The first convolutional layer has 27 filter kernels of size 8x8x3; the second has 90 kernels of size 7x7x27 and the last has 180, 5x5x90 filter kernels. After the third convolutional layer and the first fully connected one a dropout is performed during training with a 50% rate. The first fully connected layer has 420 units, and the last 9, corresponding to the number of classes. Finally, the training for classification is performed with a 9-way softmax.

The architecture and used regularizations for training has been selected through experimentation but also following the results for improving performance of CNNs described in [16].

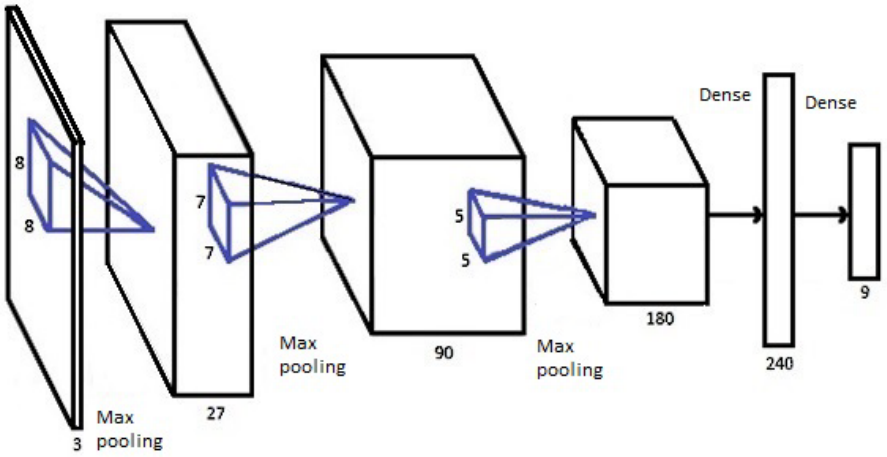


Fig. 3. Illustration of the architecture used for the CNN

Table 1. Confusion matrix of the network for the nine classes tested on the validation set

	Build-ings	Sky	Plants	Interior	Exterior	Floor	Far C.	Mid-dist.	Crowd
Buildings	**0,70**	0,01	0,02	0,14	0,07	0,02	0,00	0,02	0,03
Sky	0,00	**0,96**	0,01	0,01	0,00	0,01	0,00	0,00	0,01
Plants	0,02	0,02	**0,72**	0,00	0,08	0,08	0,02	0,02	0,04
Interior	0,03	0,02	0,00	**0,91**	0,00	0,02	0,00	0,01	0,00
Exterior	0,20	0,01	0,04	0,04	**0,49**	0,06	0,05	0,02	0,09
Floor	0,03	0,05	0,02	0,04	0,02	**0,83**	0,00	0,00	0,01
Far C.	0,00	0,00	0,01	0,00	0,01	0,00	**0,85**	0,12	0,01
Mid-dist.	0,01	0,00	0,00	0,01	0,01	0,00	0,12	**0,77**	0,09
Crowd	0,02	0,00	0,00	0,00	0,00	0,00	0,01	0,18	**0,79**

The network has been trained by using the MatConvNet implementation [17].

In this way, for classification a sliding window extracts patches from the frame and uses the classification delivered by the network to determine the presence of one of the three class of crowd in different areas of the image.

4 Results

With the selected architecture and training on the dataset for 200 iterations, the results indicate that the problem of segmenting crowds in FPV videos can definitely be addressed as an image classification task with features learnt at the crowd level. The classification of the three possible crowd's classes with different distances from the camera shows a good performance and allows the

Fig. 4. Examples of crowd detection, the three classes are represented with different colors, from Red for crowds close to the camera, "Crowd" class, to "Far crowd" in blue. In a, the original frames and in b the respective visualization of the crowd's detection

Table 2. Confusion matrix for the classification of crowds and background in general

	Background	Crowd
Background	0.95	0.05
Crowd	0.03	0.97

mapping of crowds around the wearer just as intended. In Fig. 4 are visualized the classifications of two particular situations form a video in the data set presented in [4]. In these examples the classification of the three possible distances from the camera are illustrated.

Table 1 shows the confusion matrix of the classification performed by the selected CNN. It can be seen that the three classes for crowds have a classification rate of at least 77%, yet most part of the error in these classes are misclassifications between themselves. Thus, if the accuracy is measured as the classification of crowds and background in general, it goes up to 97% as shown in Table 2.

Even if the classification rate is quite high, the misclassification errors can generate noise, which would accumulate when data are used for tracking or mapping of the surroundings. This problem is beyond the scope of this work but could certainly be addressed by the correlation of data in a temporal framework, taking into account the information of consecutive frames in the video, and probably additional information form the ego-motion.

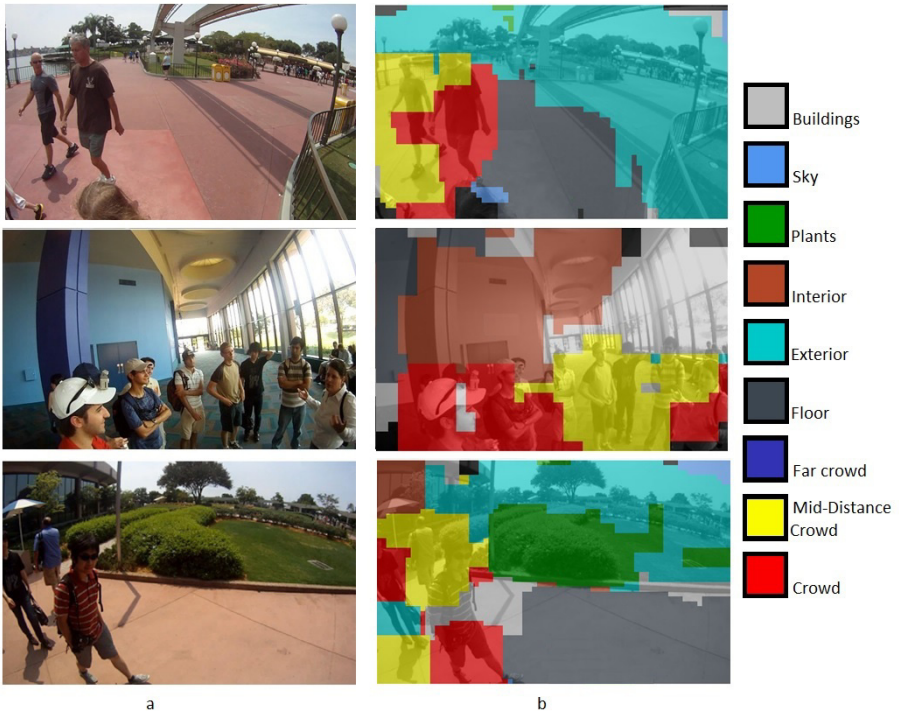


Fig. 5. Examples of segmentation of images based on the classification performed for background and crowds

Additionally to the detection of crowds, it is shown in Fig. 5 that as the background has been grouped in separated classes, though not with the same accuracy than for crowds in all of them, such division can be used to segment images as a basis for understanding context. It can be appreciated in the illustrations of Fig. 5 that, for example, it might be possible to infer the difference between a scenario inside a building where the classes "Interior" and "Buildings" are predominant on an area of the image, while in the scenes outdoors the class "Exterior" is predominant, and for example some areas where plants are present appear clearly segmented giving more hints about the context.

5 Discussion and Future Work

This work discussed how detection of people and crowds from first-person view can be achieved with good accuracy by learning features at the crowd level for classification through Convolutional neural networks.

It has been also shown that this kind of approach could be extended to wider segmentation of the image in order to generate relevant information for context awareness or other applications. This approach could be the basis for

more abstract processing that deals with interaction understanding or dynamic adjustment of the system depending on the environment.

Finally, once the detection of the crowds has been performed by using the three defined classes, it could be possible to map the crowds by projecting the angle and depth, assuming the angle as a linear function of the horizontal position of the crowd and the depth depending on the class it belongs to. This can be the next step to bring the temporal information of a video to the stabilization of crowd's detection and the extraction of information about interactions of the wearer with the surrounding people.

References

1. Betancourt, A., Morerio, P., Regazzoni, C., Rauterberg, M.: The evolution of first person vision methods: A survey. IEEE Transactions on Circuits and Systems for Video Technology **PP**(99), 1–1 (2015)
2. Bourdev, L., Yang, F., Fergus, R.: Deep poselets for human detection (2014). arXiv preprint arXiv:1407.0717
3. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 407–414. IEEE (2011)
4. Fathi, A., Hodgins, J.K., Rehg, J.M.: Social interactions: a first-person perspective. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1226–1233. IEEE (2012)
5. Kanade, T., Hebert, M.: First-person vision. Proceedings of the IEEE **100**(8), 2442–2453 (2012)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
7. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 253–256, May 2010
8. Lee, Y.J., Grauman, K.: Predicting important objects for egocentric video summarization. International Journal of Computer Vision, 1–18 (2014)
9. Narayan, S., Kankanhalli, M.S., Ramakrishnan, K.R.: Action and interaction recognition in first-person videos. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 526–532. IEEE (2014)
10. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 2056–2063. IEEE (2013)
11. Pirsiavash, H., Ramanan, D.: Detecting activities of daily living in first-person camera views. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2847–2854. IEEE (2012)
12. Poleg, Y., Arora, C., Peleg, S.: Temporal segmentation of egocentric videos. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2537–2544. IEEE (2014)

13. Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 512–519, June 2014
14. Ryoo, M.S., Matthies, L.: First-person activity recognition: what are they doing to me? In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2730–2737. IEEE (2013)
15. Sermanet, P., Kavukcuoglu, K., Chintala, S., LeCun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3626–3633. IEEE (2013)
16. Smirnov, E.A., Timoshenko, D.M., Andrianov, S.N.: Comparison of regularization methods for imagenet classification with deep convolutional neural networks. AASRI Procedia **6**, 89–94 (2014)
17. Vedaldi, A., Lenc, K.: Matconvnet-convolutional neural networks for matlab (2014). arXiv preprint arXiv:1412.4564

E-COmate: What's Your Non-consumption?

Veranika Lim[1,2(✉)], Mathias Funk[1], Matthias Rauterberg[1], Lucio Marcenaro[2], and Carlo Regazzoni[2]

[1] Designed Intelligence Group, Department of Industrial Design, Eindhoven University of Technology, Eindhoven, The Netherlands
{v.lim,m.funk,g.m.w.rauterberg}@tue.com
http://www.springer.com/lncs
[2] Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture, University of Genoa, Genoa, Italy
{lucio.marcenaro,carlo.regazzoni}@unige.it

Abstract. Most people lack awareness and hence understanding about how food-related behavior affects the environment. This commonly results in unsustainable food-related decision making, such as food waste. We propose E-COmate, an augmented bin that measures the weight of food waste through a USB postal scale, a bin and a Raspberry Pi with a Wi-Fi module, and give direct feedback to its users through a tablet by visualizing metaphorical units of the weighted food waste. We intend explore the use of E-COmate in redirecting behavior through transparency, visibility and social influence strategies like social comparison. In this paper, we present our concept, implementation, design rationale and plan of research which we expect to provide insights into the potential of eco-feedback integrated in smart home technology for food sustainability.

Keywords: Eco-feedback · Food sustainability · Social interaction

1 Introduction

The lack of awareness of everyday food behaviors goes hand in hand with a lack of understanding about its consequences for our environment [1]. Increasing awareness around food waste is one of the main approaches to reduce environmental impacts of food related behavior [1]. Roughly one-third of food produced for human consumption is lost or wasted globally, which accounts to about 1.3 billion tons per year [2]. A commonly used strategy to increase awareness of resource use and to encourage conservation is through eco-feedback [3]. The aim of eco-feedback is to increase awareness by automatically sensing peoples' activities and feeding related information back through computerized means, to foster positive attitudes towards sustainability [4] and hence the adoption of sustainable behaviors. It replaces hidden environmental information and behavior

V. Lim—This work is supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments (ICE), which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 486–499, 2015.
DOI: 10.1007/978-3-319-19258-1_40

patterns with more accessible and understandable information [5]. To our knowledge, little attention has been paid on how to apply eco-feedback on reducing household food waste with the prospective to increase awareness and lower the impact of every day food-related decision-making on the environment. In this paper we present E-COmate, an eco-feedback system that measures (figure 1) and visualizes household food waste in metaphorical units and provide a comparison with the performance of other households (figure 2). E-COmate is based on our previous prototype described in [6], which was used for an initial pilot test of a week. The previous version was limited in visualization as it could only show wasted meals for a period of a week. With our design rationale and research aims we intend to explore the effects of social influence strategies on food-related decision-making and associated social aspects.

Fig. 1. The bin-scale system of E-COmate with wooden enclosure to protect the Raspberry Pi and to keep the scale and bin fixed. Led lamps indicate when the Raspberry Pi with its program has started (red), when the scale is connected to the Raspberry Pi (orange) and when the Raspberry Pi measured a stable weight and is uploading it to a server.

2 Background

Food technology in homes is expected to have a major impact on the sustainability of food consumption practices. Smart refrigerators, for example, are capable to record expiration dates, so that food can be closely monitored and prevented to get wasted [7]. Cooking appliances are being designed with embedded interactive games or technology to increase interests on the food waste

problem [8,9] or to promote change towards alternative meals [10,11]. Although eco-feedback is generally used for other types of consumption such as energy or water use in households [12], its impacts on food waste and related behavior is rarely studied. With current concerns on food security and the global issue on food waste we propose to explore how eco-feedback can be useful for adopting sustainable food practices.

Visibility on Meaning. Throughout all food-related practices we use a crucial function of our brain; the ability to use stored information (what food we have at home) to imagine and predict possible future events such as what to buy or what to cook and eat [13]. Goal and future-oriented behavior and our ability to predict correctly is crucial in whether we will waste or not. When having a busy lifestyle, food practices receive less attention and behaviors resulting into food waste become more unconscious [14]. Eco-feedback could allow consumers to learn, 'incidentally' and without the requirement of conscious effort, what it means to waste food. For example, when we dispose food, unperceivable sources are disposed at the same time which we cannot see as it is beyond our immediate perception. Water and energy that was used for production are all lost when food is disposed. Next to natural resources we also waste physical energy, time and money. With our human ability to imagine non-perceivable or non-existent situations, eco-feedback could help our imagination by visualizing a future prediction: such as if current unsustainable behavior is continued (e.g. we would need 2 earths to produce food, considering one third is wasted) or present a 'what if' situation (e.g. if I didn't throw away this amount of food, I could have had x amount of free food). In sustainability research, information-based strategies were found to be effective at reducing overall energy usage in controlled experimental studies which indicates the potential of educational information technologies targeting conservation in households [15].

3 Design Probe and Rationale

E-COmate is an augmented bin that measures the weight of food waste and give direct feedback to its users through an Android application by visualizing the potential number of servings that has been wasted (See figure 1 and 2). The prototype consists of a laser-cut wooden enclosure with a USB postal scale, a bin and a Raspberry Pi with a Wi-Fi module. The application runs on a tablet computer to visualize the amount of wasted potential servings. With the visualization we aim at providing meaning of wasted food to increase awareness of hidden facts. Visualizations could be done through the number of bottled water, landfills or calories used for the production of the food that is wasted, the number of people that could have been fed for a day or how many earths we need if everyone would continue wasting the same. However, we chose to visualize the number of servings lost because it is directly linked to consumers own daily concerns, wasted meals equals potential free meals. Through real-time

feedback embedded in their home environment, we provide a way for consumers to learn and reflect about their everyday behaviors on a daily basis.

Metaphors to Improve Understanding. In general, eco-feedback research use metaphors for visualization to enhance understanding of the information that is fed back to the user. Researchers have explored ways to inform consumption by using metaphorical units instead of volumetric units; such as everyday objects like the number of jugs and oil trucks for water usage instead of just the visualization of the number of gallons or liters [16]. In another recent example researchers used carbon weight to indicate environmental health visualized on a bathroom scale [12]. E-COmate visualizes the number of potential servings as a metaphorical unit converted from the weight of wasted food. We expect that the number of potential servings could be associated with monetary loss and hence another means for persuasion. Although studies on energy use have pointed out that financial incentives have short-lived effects and can be counterproductive [15], whether it also applies to food waste is unclear. Financial benefits from saving energy are often quite small compared to other household expenses like food [17].

Social Influence Strategies as a Means for Persuasion. To advance the impact of eco-feedback on food waste, we are interested in implementing social influence strategies in the design of the system and explore its effects on social aspects of food waste related behavior. We seem to compare ourselves to others to find out how we're doing when objective measures for self-evaluation are unavailable [18]. Hence, a common influence strategy in eco-feedback research is the use of social comparison as motivation for reducing consumption. Social approval and norm activation are principles that humans use to influence others and seem to be successful when facilitated through technology for supporting behavioral change towards more sustainable practices [19]. Consumers seem to be more motivated to save energy and water when they are able to compare consumption with others [20,21]. Including social aspects when studying food waste related behavior, is therefore considered critical.

3.1 Software and Hardware Design

E-COmate includes an off-the-shelf Dymo M5 USB postal scale which transmits its measurements via a USB connection to the Raspberry Pi. The Raspberry Pi handles the collection and saving of data into a server, as well as the communication with the scale. An eco-feedback manager reads the data from the server and controls the visualizations in the Android application which is programmed in JavaScript. All button presses in the application are saved in the server. All data in the server can be exported to csv or text files for analysis. Implementations are planned for 4 households in which each bin-scale system has his own id and its unique associated Android application with a similar id (see figure 3).

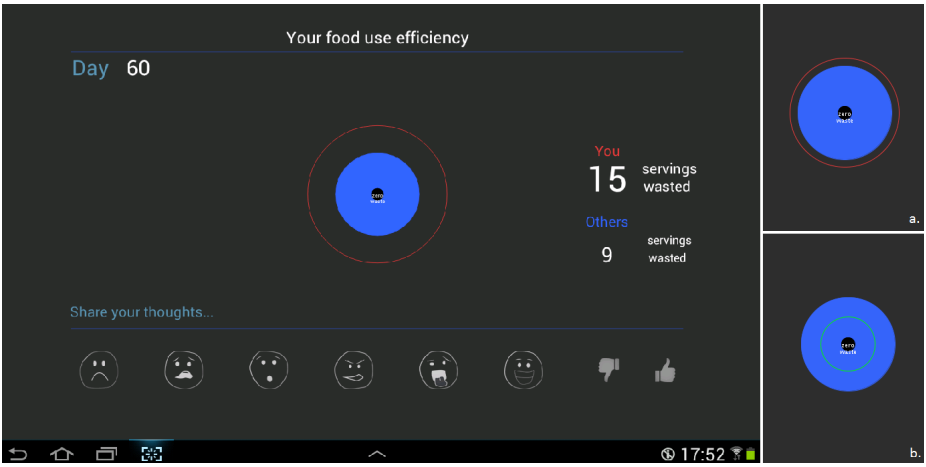


Fig. 2. The data from the bin-scale system is fed into an application for visualization of food waste on a tablet computer. This figure presents a display with social comparison with negative feedback relatively to *others*. The radius of this blue circle presents *others* and is kept constant for simplicity and consistency (*see a.*). When users waste less than *others*, "you" and the red circle changes to green and the circle then falls inside the blue circle (*see b.*)

3.2 Information Design

The display shows 2 types of information feedback (See figure 2). Numerical values are provided for the number of days and the amount of potential servings wasted by users since the start of usage. When the display includes social comparison, another number is presented for the amount of servings wasted by *others*. The numerical values are supported with a visualization based on the common visualization used for the *Earth overshoot day* concept; which signifies the gap between our demand for ecological resources and services, and how much the planet can provide [22]. Based on this idea, the display visualizes the relative difference between zero waste and participants' current waste and/or the amount others are wasting. The colors green and red are used for positive and negative feedback respectively in comparison with others. The display also includes buttons for users to interact and express feelings towards the feedback. We used six universal emotions *sadness*, *surprise*, *disgust*, *fear*, *anger* and *happiness* [23] and included a thumbs up and a thumbs down in case users do not want to express emotions.

3.3 Objective

Our main objective is to explore the effects of using eco-feedback applied to food waste. Particularly we are interested in how social comparison information can play a role in the motivation to adopt sustainable behaviors in daily food

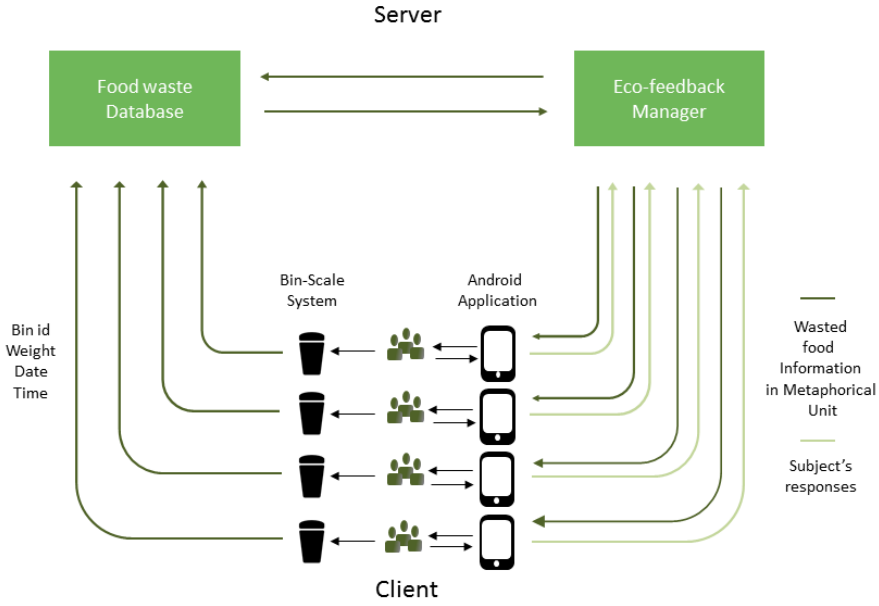


Fig. 3. Information flow of E-COmate for a study with 4 user groups

related practices. We include social aspects in our explorations as we consider it an important measure related to food practices. We investigate the following research questions:

1. How does eco-feedback affect food waste?
2. How does eco-feedback affect environmental attitude, social value orientation, memory and sustainable behavior in general?
3. How does eco-feedback with or without social comparison affect reflection and motivation?
4. How does eco-feedback with or without social comparison affect social group dynamics around food practices?
5. How should eco-feedback be designed to gain efficiency and be effective in reducing food waste?

4 Study Design

4.1 Participants

We recruited student houses that will use E-COmate for 1 month continuously including 7 days for the baseline measurement. Each house uses one E-COmate. Participants are recruited based on the following requirements; (1) they have to do groceries themselves which requires responsibility and (2) they have to eat or

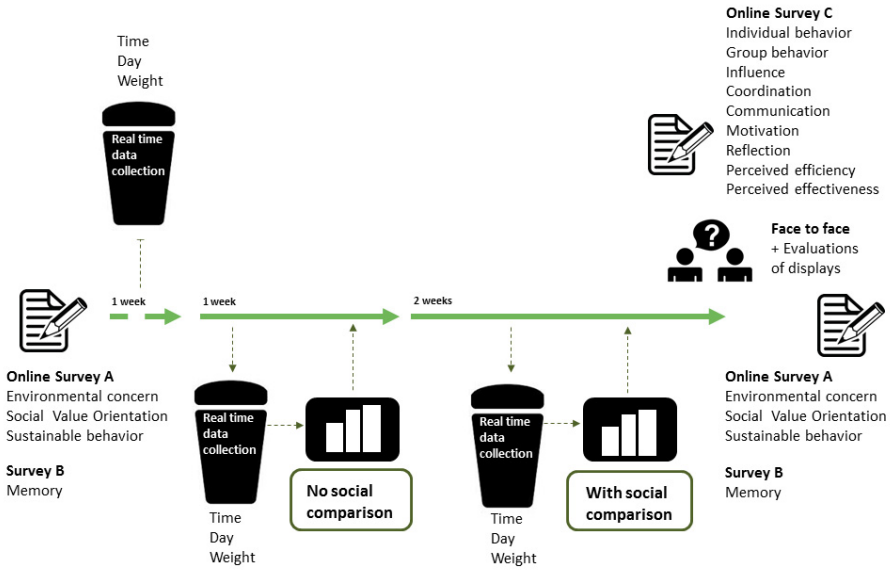


Fig. 4. Experimental design

cook at home at least three times a week which would mean that there will be ingredients stored in homes hence with a possibility of wastage.

4.2 Experimental Design

The study will start with a baseline measurement of 7 days without any interventions in terms of eco-feedback. This baseline aims at getting users used to separating food waste and using the scale-bin system. All participants will then be presented with 2 types of displays alternately. In the first week after the baseline, the application displays eco-feedback without social comparison (condition 1) followed by eco-feedback with social comparison (condition 2) for another 2 weeks. For research purposes we simulate the food waste information of *others* by randomly taking 40 % to 110 % of users actual waste. This means that users will mostly get negative feedback. This way we can look into the effects of receiving negative feedback when behavior is compared to other people or national standards. Although any food waste should be prevented anytime, social norms might have some influence. Our rationale behind this is to control for consistency among all participants in feedback provided independent from the differences in waste. Dependent measures will be gathered through the bin, online surveys and an interview (See figure 4 for an overview and time line of the experimental design).

4.3 Measuring the Effects of Eco-feedback

Two online surveys will be conducted in the beginning and at the end of the study to explore effects of eco-feedback. One survey contains three standardized questionnaires to measure ecological concern, social value orientation and general sustainable behavior. The second online survey aims at measuring participant's performance on memory.

Environmental Concern. The level in which people are concerned about their environment can be measured, among others, by the Ecological Motives Scale [24]. This 12-item scale provides a measure of someones concern about environmental problems because of the consequences that result from harming nature. Participants are asked to rate items from 1 (not important) to 7 (supreme importance) in response to the following question:

"I am concerned about environmental problems because of the consequences for _____".

The scale measures three categories of concern about environmental problems caused by human behaviors: Egoistic, Altruistic, and Biospheric. Egoistic items are: *me*, *my future*, *my lifestyle*, and *my health*. Altruistic items are: *all people*, *children*, *my children*, and *people in my country*. Biospheric items are: *plants*, *animals*, *marine life*, and *birds*. Subscales are scored by averaging the scores from the items within the subscales. Higher scores represent higher concern with a given subscale. This measure could give us insights in how people respond to using our prototype depending on their level of concern.

Social Value Orientation. Social value orientation (SVO) is a person's preference about how to allocate resources between the self and another person [25]. The measure has six primary items (See figure 5 for an example item), used to group people who seek to maximize their gains into being pro-self or competitive and people who are also concerned with other's gains and losses into being pro-social or altruistic. All of the items have the same general form, each is a resource allocation choice over a well defined continuum of joint payoffs. We believe this measure could be relevant and possibly related to food-related behavior. Food behavior is socially and culturally integrated and preventing food waste comes together with social responsibility, for future generations as well. Hence, this measure could give us insights in how people respond to using our prototype depending on their social value orientation. Not wasting food could, however, also be caused by having a self-centered reason such as saving money.

Prospective Memory. The bad performance of memory of past information has been indicated as one of the main causes in qualitative food waste studies [14]; e.g. forgetting what is available at home or the days how long an item is good for due to busy and unpredictable lifestyles. A subcategory of memory,

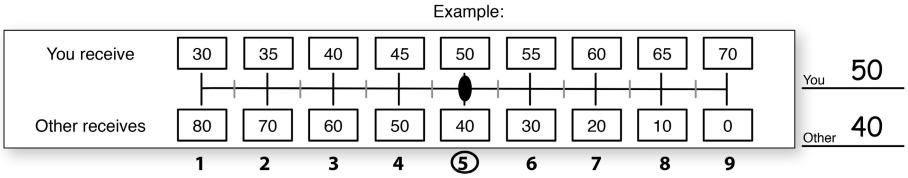


Fig. 5. An example of a primary SVO items

prospective memory, which is the memory of future events, is found to be crucially involved in our ability to imagine non-existent events and simulate future happenings [13,26]. This type of memory is obviously also related to food waste behavior; e.g. future plans on what to do with the available items at home. Both types of memory, about the past and the future is, therefore, another kind of human aspect that should be taken into account when exploring technological interventions for food waste prevention. In our study we ask participants to describe two past events and two future events related to food behavior with as much detail as possible (e.g. time, place, people etc.); a dinner from the past week, a planned dinner in the next few days, food items they had available last week for at least two days and food items they are planning on purchasing in the next few days. Answers can be scored for detail considering the number of words, number of distinguishable aspects, and the specificity of time and space.

General Ecological Behavior. Although, holding a certain attitude might affect behavior, this is not always the case. Therefore, in addition to levels of concern, attitude, values and intention triggered through memory, we should look at actual behavior. We use a questionnaire from Kaiser [27] which includes 50 items addressing a wide range of conservation behaviors. Respondents are asked to indicate whether or not they have ever engaged in a particular behavior. Topics include energy conservation, mobility and transportation, waste avoidance, consumerism, recycling, vicarious social behaviors towards conservation and in addition, we included 6 items related to food waste (See table 1). People who are raised in a sustainable environment are more likely to adopt a variety of sustainable behaviors. When one sustainable behavior is adopted, people might also be more likely in adopting other sustainable behaviors to stay consistent. Here, we build on the theory of cognitive dissonance [28], the feeling of discomfort when simultaneously holding two or more conflicting cognitions: ideas, beliefs, values or emotional reactions. In this case, behavior could lead to an adaptation of attitude and other behaviors when using sustainable technology.

4.4 The Potential of Applying Social Comparison

A third online survey will be conducted at the end to explore the effects of eco-feedback with social comparison or the lack thereof and users' preferences. Here, we are interested in individual and group related behavior, which is further

Table 1. 56 conservation behaviors grouped into seven performance domains

Table General Ecological Behavior
Energy conservation
1. I own energy efficient household devices
2. I wait until I have a full load before doing laundry
3. I wash dirty clothes without prewashing
4. In hotels, I have the towels changed daily
5. I use a clothes dryer
6. I bought solar panels to produce energy
7. I use renewable energy sources
8. In the winter, I keep the heat on so that I do not have to wear a sweater
9. In the winter, I leave the windows open for long periods of time to let in fresh air
10. In winter, I turn down the heat when I leave my apartment for more than 4 h
11. I prefer to shower rather than to take a bath
Mobility and transportation
12. I drive my car in or into the city
13. I drive on freeways at speeds under 62.5 mph
14. I keep the engine running while waiting in front of a railroad crossing or in a traffic jam
15. At red traffic lights, I keep the engine running
16. I drive to where I want to start my hikes
17. I refrain from owning a car
18. I am a member of a carpool
19. I drive in such a way as to keep my fuel consumption as low as possible
20. I own a fuel-efficient automobile (less than 7 liters per 100 km)
21. For longer journeys (more than 6 h), I take an airplane
22. In nearby areas (around 30 km), I use public transportation or ride a bike
23. I ride a bicycle or take public transportation to work or school
Waste avoidance
24. I buy milk in returnable bottles
25. If I am offered a plastic bag in a store, I take it
26. I reuse my shopping bags
27. I buy beverages in cans
28. I buy products in refillable packages
51. I tend to overbuy food that get wasted
52. Past food date is a common reason for throwing out food
53. Forgetting leftovers is a common reason for throwing out food
Consumerism
29. I use fabric softener with my laundry
30. I use an oven cleansing spray to clean my oven
31. I kill insects with a chemical insecticide
32. I use a chemical air freshener in my bathroom
33. I buy convenience food
34. I buy seasonal products
35. I buy bleaches and colored toilet paper
36. I buy meat and products with eco-labels
37. I buy domestically grown wooden furniture
54. I am a vegetarian
55. Sometimes I don't mind eating vegetarian
56. I want to reduce my meat consumption
Recycling
38. I collect and recycle used paper
39. I bring empty bottles to a recycling bin
40. I put dead batteries in the garbage
41. After meals, I dispose of leftovers in the toilet
Vicarious, social behaviors towards conservation
42. After a picnic, I leave the place clean as it was originally
43. I am a member of an environmental organization
44. I read about environmental issues
45. I contribute financially to environmental organizations
46. I talk with friends about problems related to the environment
47. I have pointed out unecological behavior to someone
48. I boycott companies with an unecological background
49. I have already looked into pros and cons of having a private source of solar power
50. I requested an estimate on having solar power installed

specified in planning, purchasing, preparation, and dealing with leftovers. Other items focus on how much the system supports reflection, affect motivation to change behavior and how much the system affect social interaction in terms of influence, coordination and communication. Answer options ranged from *very much* to *not at all*, with item's score contribution from 5 to 1, respectively. Semi-structured interviews at the end of the study are based on the same questionnaire items, but aiming for more in-depth discussions on *how* the system had affected their behaviors. Here, we explore perceived efficiency and effectiveness of E-COmate and gather evaluations of alternative displays and design dimensions.

4.5 Procedures

We provide the following task descriptions to participants: (1) related to the use of the bin, they will be asked to only throw away food waste that was once edible but now turned bad or currently doubtful (e.g. past due date). A list will be provided with items that are allowed (e.g. potato skins, vegetable peels and greens, bread crusts, over baked or overcooked food, plate scraps etc.) and items that are *not* allowed in E-COmate (e.g. bones, tea bags, coffee grounds, egg shells, banana peels etc.). Instructions will be provided on the procedures of using the bin and how to read and use the display. To dispose food, users have to wait until the green light goes on *and* off again before disposing something else *or* before emptying it as it needs 8 seconds to stabilize and send data to the server. Users also have to turn off the scale and wait until the orange light goes off before emptying the bin. Instructions also applies for when the system freezes; (2) related to the Android application, whenever participants see the eco-feedback on the tablet they can interact with it by indicating current emotions or whether they like it or not; (3) for data gathering other than the food waste weight, participants fill in two online survey before and three after the study. All surveys are send by email and an interview takes place at the end of the study; finally (4), comments or pictures taken by participants throughout the study can be send and shared within a WhatsApp chat group of the users belonging to one house. This chat group can also be used for any question related to the study.

5 Discussion

With E-COmate we aim at exploring the effect of using social comparison in eco-feedback on food-related behavior in individuals as well as groups of users in terms of influence, coordination and communication among users. We are further interested in how it supports reflection and affect motivation to change behavior. Overall, we are interested in exploring the effects of E-COmate on memory, attitude and sustainable behavior in general and we aim at discovering hidden temporal patterns in the data collected through E-COmate: when are people more likely to waste (e.g. in the evening or before shopping, etc.)? This would help determine the causes of food waste, currently under explored.

Quantitative and qualitative findings on these aforementioned behavioral and cognitive aspects could help in designing a smart home system that is likely to be effective for reducing food waste in households. Targeting these goals we face the following challenges: the raw data from the scale-bin system include noise such as fluctuations in the weight that is measured when nothing should change and 0 measurements when the system is turned off (e.g. when frozen or internet connection is lost). From this raw data, useful data is filtered out and converted into something understandable, meaningful and realistic for visualizations to be useful. The definition of noise, however, is based on our own assumptions (e.g. a bin will only be emptied when it has a significant content so a decrement of weight that is less than a certain value can be considered noise and an increment that is less than a certain value might be other than actual food waste such as touching the bin). Another challenge is whether an in the wild approach can reliably capture food and/or waste data at the consumer level for a period of a month. The accuracy of the electronics inside the scale can reduce by time and remotely monitoring users could limit data. Furthermore, food waste monitoring have implications on privacy matters. It is another source of peoples private behavior they might likely not want to share. Food waste information could be a target for taxes for example. This leads to the question on why people would want to use it. We, however, believe that E-COmate could lead to being more economical and be used for educational purposes and hence to increase awareness.

5.1 Future Work

Future work involves the deployment of E-COmate in multiple households for a period of a month of which findings will be used as a source of inspiration for next prototypes. In these follow up prototypes we could explore the use of different metaphors and distinguish between food types (e.g. fruits and vegetables, meat and fish, dairy, and grains). Meat for example requires much more resources, so eating as well as wasting it has much more impact on our environment. We could explore how this information can be presented and how it effects consumption or wastage. We could also look into the effects of receiving positive feedback when behavior is compared to other people or national standards (e.g. by randomly taking 90 % to 160 % of users actual waste).

Acknowledgments. We greatly thank Erik Pietersma and Geert van den Boomen for the technical support in developing the bin-scale system. We also thank Claudio Martella for his invaluable input on the application side as well his comments on the display design.

References

1. Moomaw, W., Griffin, T., Kurczak, K., Lomax, J.: The critical role of global food consumption patterns in achieving sustainable food systems and food for all. Tech Report, United Nations Environment Programme (2012)

2. Gustavsson, J., Cederberg, C., Sonesson, U., van Otterdijk, R., Meybeck, A.: Global food losses and food waste. Tech Report, Food and Agriculture Organization of the United Nations (2011)
3. Froehlich, J., Findlater, L., and Landay, J.: The design of eco-feedback technology. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1999–2008. ACM, New York (2010)
4. Pierce, J., Odom, W., and Blevis, E.: Energy aware dwelling: A critical survey of interaction design for eco-visualizations. In: Proceedings of the 20th Australasian Conference on Computer-Human Interaction, pp. 1–8. ACM, New York (2008)
5. Holmes, T.: Eco-visualization: Combining art and technology to reduce energy consumption. In: Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition, pp. 153–162. ACM, New York (2007)
6. Lim, V., Jense, A., Janmaat, J., Funk, M.: Eco-feedback for non-consumption. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 99–102. ACM, New York (2014)
7. Rouillard, J.: The Pervasive Fridge; A smart computer system against uneaten food loss. In: The Seventh International Conference on Systems, pp. 135–140. Saint-Gilles, Runion (2012)
8. Comber, R., Hoonhout, J., van Halteren, A., Moynihan, P., Olivier, P.: Food practices as situated action: exploring and designing for everyday food practices with households. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2457–2466. ACM, New York (2013)
9. Lepe Salazar, F., Yamabe, T., Alexandrova, T., Liu, Y., Nakajima, T.: Family interaction for responsible natural resource consumption. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2105–2110. ACM, New York (2012)
10. Clear, A. K., Hazas, M., Morley, J., Friday, A., Bates, O.: Domestic food and sustainable design: a study of university student cooking and its impacts. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2447–2456. ACM, New York (2013)
11. Kadomura, A., Li, C-Y., Tsukada, K., Chu, H-H., Siio, I.: Persuasive technology to improve eating behavior using a sensor-embedded fork. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 319–329. ACM, New York (2014)
12. Kuo, P-Y., Horm, M.S.: Energy diet: energy feedback on a bathroom scale. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 435–446. ACM, New York (2014)
13. Schacter, D.L., Addis, D.R., Buckner, R.L.: Remembering the past to imagine the future: the prospective brain. Nature Reviews Neuroscience **8**, 657–661 (2007)
14. Ganglbauer, E., Fitzpatrick, G., Molzer, G.: Creating visibility: understanding the design space for food waste. In: Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia, pp. 0–9. ACM, New York (2012)
15. Delmas, M., Fischlein, M., Asensio, O.: Information Strategies and Energy Conservation Behavior: A Meta-analysis of Experimental Studies from 1975–2011. Working Paper, Institute of the Environment and Sustainability, UCLA, Los Angeles (2013)
16. Froehlich, J., Patel, S., Landay, J., Findlater, L., Ostergren, M., Ramanathan, S., Peterson, J., Wragg, I., Larson, E., Fu, F., Bai, M.: The design and evaluation of prototype eco-feedback displays for fixture-level water usage data. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2367–2376. ACM, New York (2012)

17. Wolak, F.: Do Residential Customers Respond to Hourly Prices? Evidence From A Dynamic Pricing Experiment. American Economic Review **101**(3), 83–87 (2011)
18. Festinger, L.: A theory of social comparison processes. Human Relations **7**(2), 117–140 (1954)
19. Ham, J.R.C. & Midden, C.J.H. (2014).: A persuasive robot to stimulate energy conservation: the influence of positive and negative social feedback and task similarity on energy consumption behavior. International Journal of Social Robotics 6(2), 163–171 (2014)
20. Foster, D., Lawson, S., Blythe, M., and Cairns, P.: Wattsup?: Motivating reductions in domestic energy consumption using social networks. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp. 178–187. ACM, New York (2010)
21. Odom, W., Pierce, J., and Roedl, D.: Social incentive & eco-visualization displays: toward persuading greater change in dormitory communities. In: Workshop Proceedings of Public and Situated Displays to Support Communities. ACM, New York (2008)
22. Global Foodprint Network; Advancing the Science of Sustainability. http://www.footprintnetwork.org
23. Black, M.J., Yacoob, Y.: Recognizing facial expressions in image sequences using local parameterized models of image motion. Int. Journal of Computer Vision **25**(1), 23–48 (1997)
24. Schultz, P.W.: The structure of environmental concern: Concern for the self, other people, and the biosphere. Journal of Environmental Psychology **21**, 327–339 (2001)
25. Murphy, R.O., Ackermann, K.A., Handgraaf, M.J.J.: Measuring Social Value Orientation. Judgment and Decision Making **6**(8), 771–781 (2011)
26. Suddendorf, T., Busby, J.: Making decisions with the future in mind: Developmental and comparative identification of mental time travel. Learning and Motivation **36**, 110–125 (2005)
27. Kaiser, F.G., Wilson, M.: Goal-directed conservation behavior: The specific composition of a general performance. Personality and Individual Differences **36**, 1531–1544 (2013)
28. Festinger, L.: A Theory of Cognitive Dissonance. Stanford University Press, California (1957)

Mathematical and Theoretical Methods in Fuzzy Systems

Extended Bag of Visual Words for Face Detection

Gholam Ali Montazer[1(✉)], Mohammad Ali Soltanshahi[3], and Davar Giveki[2]

[1] Department of Information Technology Engineering, School of Engineering,
Tarbiat Modares University, Tehran, Iran
montazer@modares.ac.ir
[2] Department of Computer Science, University of Tehran, Tehran, Iran
Giveki@students.irandoc.ac.ir
[3] Department of Information Technology,
Iranian Research Institute for Information Science and Technology (Iran Doc), Tehran, Iran
ali.soltanshahi@gmail.com

Abstract. Face detection shows a challenging problem in the field of image analysis and computer vision and therefore it has received a great deal of attention over the last few years because of its many applications in various areas. In this paper we propose a new method for face detection using an Extended version of Bag of Visual Words (EBoVW). Two extensions of the original bag of visual words are made in this paper, fist, using Fuzzy C-means instead of K-means clustering and second is, building histogram of words using multiple dictionaries for each image. The performances of the original BoVW model with K-means and the proposed EBoVW are evaluated in terms of Area Under the Curve (AUC) and Equal Error Rate (EER) on MIT CBCL Face dataset which is a very large face dataset. The experimental results show the proposed model achieves very promising results.

Keywords: Face detection · Bag of visual words · Extended bag of visual words · Scene classification · SIFT · Fuzzy C-means

1 Introduction

Biometric-based techniques have been born as the most promising option for recognizing individuals in recent years since, instead of authenticating people and granting them access to physical and virtual domains based on passwords, PINs, smart cards, plastic cards, tokens, keys and so forth, these methods examine an individual's physiological and/or behavioral characteristics in order to determine and/or ascertain his identity. Passwords and PINs are hard to remember and can be stolen or guessed; cards, tokens, keys and the like can be misplaced, forgotten, purloined or duplicated; magnetic cards can become corrupted and unreadable. However, an individual's biological traits cannot be misplaced, forgotten, stolen or forged. Face detection appears to offer several advantages over other biometric methods, a few of which are mentioned here: almost all these technologies require some voluntary action by the user, i.e., the user needs to place his hand on a hand-rest for fingerprinting or hand geometry detection and has to stand in a

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 503–510, 2015.
DOI: 10.1007/978-3-319-19258-1_41

fixed position in front of a camera for iris or retina identification. However, face detection can be done passively without any explicit action or participation on the part of the user since face images can be acquired from a distance by a camera. This is particularly beneficial for security and surveillance purposes. Furthermore, data acquisition in general is fraught with problems for other biometrics: techniques that rely on hands and fingers can be rendered useless if the epidermis tissue is damaged in some way (i.e., bruised or cracked). Good face detection algorithms and appropriate preprocessing of the images can compensate for noise and slight variations in orientation, scale and illumination. Face detection techniques can be broadly divided into three categories based on the face data acquisition methodology: methods that operate on intensity images; those that deal with video sequences; and those that require other sensory data such as 3D information or infra-red imagery [1]. In this paper we focus on face detection from Intensity Images which fall into two main categories: feature based and holistic [2-4]. Feature based approaches first process the input image to identify and extract (and measure) distinctive facial features such as the eyes, mouth, nose, etc. and then compute the geometric relationships among those facial points, thus reducing the input facial image to a vector of geometric features. Standard statistical pattern detection techniques are then employed to match faces using these measurements. Holistic approaches attempt to identify faces using global representations, i.e., descriptions based on the entire image at once rather than on local features of the face [1].

Here, we propose a new face detection system using an extended version of Bag of Visual Words (BoVW). Object detection using BoVW model is one of the most acceptable methods for object classification. The BoVW approach has shown acceptable performance because of its fast run time and low storage requirements [5-9]. The key idea is to quantize each extracted key point into one of the visual words, and then represent each image by a histogram of the visual words. To this end, different clustering algorithms like K-means are generally used for generating the visual words. Constructing the BoVW from the images involves the following steps: (1) Automatically detect regions/points of interest (local patches) and compute local descriptors over these regions/points using some image descriptors, (2) quantize the descriptors into words to form the visual codebook using some clustering techniques, (3) find the occurrences in the image of each specific word in the vocabulary in order to build the BoVW features namely histogram of words and finally (4) Learn the histogram of words using a classifier like SVM, ANN or Adaboost. In this work we implemented BoVW method using dens version of the Scale Invariant Feature Transform descriptor (SIFT) [10]. After obtaining local features called descriptors, a codebook is generated to represent them. The codebook is a group of codes usually obtained by clustering over all descriptors. Typically there are two forms of clustering; hard clustering and soft clustering. The performance of BoVW highly depends on some factors such as the dictionary generation method, size of dictionary, similar words detection, meaningless worlds elimination, normalization, similarity measure and classifier. This paper investigates a new method of generating the dictionary of visual words in which multiple dictionaries are used for constructing BoVW using Fuzzy C-means, this approach uses more visual words than original BoVW. In this way more distinctive image features can be captured in a semantic manner. The resulting distribution of descriptors is quantified by using vector quantization to form histogram of words.

Compared to the original BoVW our proposed method considerably increases the performance of the algorithm for large scale collection of face images of MIT CBCL dataset with 31022 images.

Regarding improving, extending or modifying feature extraction methods that mostly use SIFT features, some notable works have been done. For instance author in [11], presented a novel framework for detecting, localizing, and classifying faces in terms of visual traits, e.g., sex or age, from arbitrary viewpoints and in the presence of occlusion. All three tasks were embedded in a general viewpoint-invariant model of object class appearance derived from local scale-invariant features, where features were probabilistically quantified in terms of their occurrence, appearance, geometry, and association with visual traits of interest. An appearance model was first learned for the object class, after which a Bayesian classifier was trained to identify the model features indicative of visual traits. The proposed framework can be applied in realistic scenarios in the presence of viewpoint changes and partial occlusion, unlike other techniques assuming data that are single viewpoint, upright, prealigned, and cropped from background distraction.

The rest of the paper is organized as follows. Section 2 discusses the Extended Bag of Visual Words (EBoVW) model. In section 3 experimental results are reported. Finally conclusions are drawn in section 4.

2 Extended Bag of Visual Words

In the original bag of visual words model single dictionary is generated from stack of features. In our proposed extended bag of visual words (EBoVW) multiple dictionaries for each image are constructed so that we can have more visual words and hence EBoVW has significantly increases the performance for large scale image classification problem. Fig. 1 shows the steps of proposed EBoVW model. To this end each dictionary Dic_N is built with a different subset of the image features. So in the implementation of the EBoVW for generating multiple dictionaries the image gets a histogram $hist_N$ from every dictionary Dic_N which is concatenated to form a single histogram hist. Every feature gets N entries in the histogram hist, one from every dictionary so that more different words are taken from different independent dictionaries while in the original BoVW method more words are taken from the same dictionary. Thus the EBoVW method not only demands less storage than the original method but also its features are more distinctive.

For implementing the proposed EBoVW method first N random possibly overlapping subsets of the image features $\{S_n \ , n = 1, \dots, N\}$ are extracted. Afterwards, a dictionary Dic_n is computed independently for each subset S_n. Each dictionary has a set of V_n visual words. Then the histogram is computed. Every image feature gets its visual word from every dictionary Dic_n. Accumulate these visual words as individual words into individual histograms $hist_n$ for each dictionary. The final histogram is the concatenation of the individual histograms. In this work, we used Fuzzy C-means clustering for implementing the EBoVW model . Fuzzy C-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. The advantage of soft clustering is that it is insensitive to noise. In many real situations, fuzzy clustering is more natural than hard clustering, as objects on the boundaries

between several classes are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial memberships. The steps for EBoVW using fuzzy clustering is shown in Fig. 1 From the stack of features N subsets of features are taken randomly and N dictionaries are constructed using Fuzzy C- means clustering. For each of the constructed dictionaries, histograms are generated for each image in the dataset and the final histogram is the concatenation of the individual histograms. This is done during the training phase of the algorithm. During the testing phase features are extracted from each image and histogram for the image is generated by the same process as stated above. The SVM classifier then predicts the class of the query images.

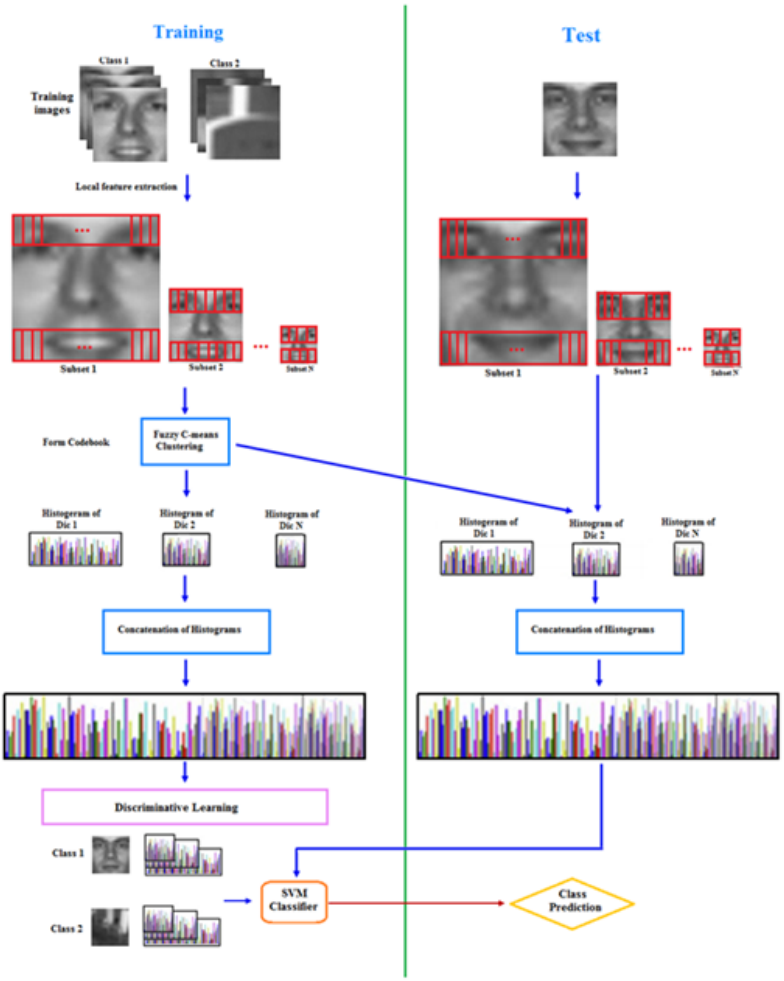


Fig. 1. Steps for the Extended Bag of Visual Words construction

3 Experimental Results

We present the results of face detection experiments over the images of the MIT CBCL [12]. This dataset consists of 31,022 grayscale images of 2,429 faces and 4,548 non-faces for training and 472 faces and 23,573 non-faces for testing. All of the images with a size of 19×19 pixels. The face images have various illumination conditions, facial expressions (e.g., open/closed eyes, smiling/not smiling) and facial details (e.g., glasses/no glasses).

The Bag of visual words model for face detection was implemented using Dense SIFT. 128 dimensional feature vectors used for representing the extracted features. The extracted features are clustered using Fuzzy C-means clustering algorithm and a codebook is generated with each vector in it being a visual word which serves as the basis for indexing the images. Images are then represented as histogram counts of these visual words. For the SVM classifier an exponential kernel of the form $\exp -\alpha d$ was applied, where d is the Euclidean distance between the vectors, and the scalar α is determined as described in [13] (we use the LIBSVM package [14] with the trade-off between training error and margin at $C = 1$).

Table 1. Performance comparison (AUC) between original BoVW and EBoVW

Length of feature vector	Original BoVW	EBoVW with 2 Dictionaries	EBoVW with 3 Dictionaries	EBoVW with 4 Dictionaries
128	92.08%	94.05%	94.92%	96.36%
256	**92.16%**	95.64%	95.87%	**96.71 %**
512	91.64%	94.67%	95.56%	96.44%
1024	90.60%	94.88%	95.52%	96.54%

Table 2. Performance comparison (EER) between original BoVW and EBoVW

No Of Words Per Dictionary	Original BoVW	EBoVW with 2 Dictionaries	EBoVW with 3 Dictionaries	EBoVW with 4 Dictionaries
128	14.44%	13.06%	12.09%	10.67%
256	16.53%	12.71%	11.86%	9.96%
512	15.68%	13.11%	11.44%	10.78%
1024	**14.19%**	11.13%	10.81%	**9.32%**

In order to evaluate the efficiency of the proposed method two criteria have been selected from the state-of-the art [15]. Receiver Operating Characteristic (ROC) curve and Equal Error Rate (EER). ROC curve is used to summarize the performance of a biometric verification system. The ROC curve plots, parametrically as a function of the decision threshold, the percentage of impostor attempts accepted (i.e. true positive rate (TPR)) on the x-axis, against the percentage of genuine attempts accepted (i.e. 1 - false positive rate (FPR)) on the y-axis. The ROC curve is threshold independent, allowing performance comparison of different systems under similar conditions. The Area Under the Curve (AUC) is an indicator of the overall quality of a ROC curve. The ROC of the ideal classifier has AUC equal to 1.

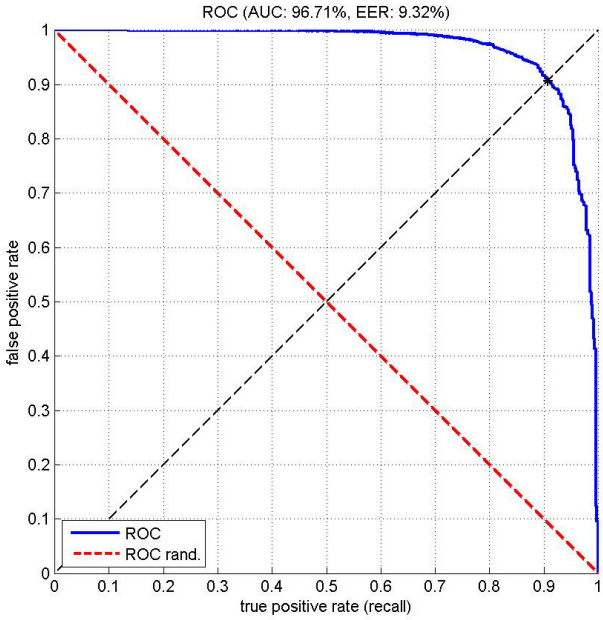


Fig. 2. ROC curve of BoVW method

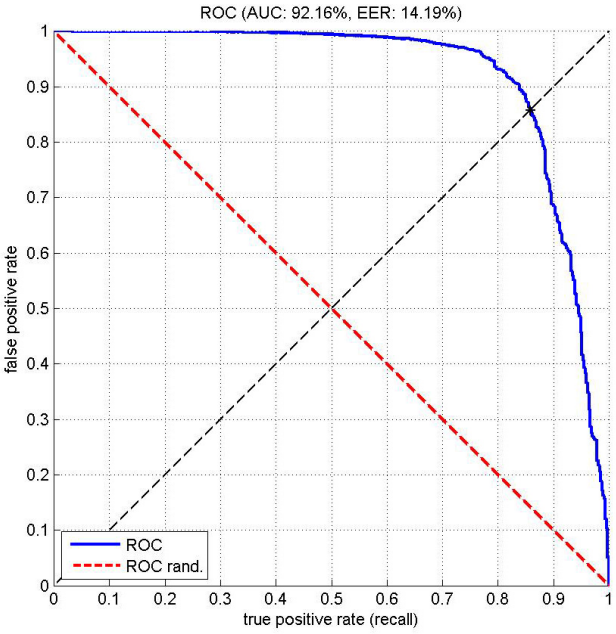


Fig. 3. ROC curve of BoVW method

Another indicator is the Equal Error Rate (EER), the point on the ROC curve that corresponds to have an equal probability of miss-classifying a positive or negative sample. This point is obtained by intersecting the ROC curve with a diagonal of the unit square. The AUC is an overall summary of detection accuracy. AUC equals 0.5 when the ROC curve corresponds to random chance and 1.0 for perfect accuracy. On rare occasions, the estimated AUC is <0.5, indicating that the test does worse than chance. In general, the larger the AUC value, the higher the accuracy of the biometric system. Also, the lower the equal error rate value, the higher the accuracy of the biometric system. In our experiments we computed AUC and EER for each experiment as it is shown in Tables 1 and 2. From these tables compared to the original BoVW model, EBoVW model achieves higher performance in terms of both AUC and EER. The best results in both methods were achieved with feature vector of length 1024. In the original BoVW model the best AUC and the best EER are 92.16% and 14.19% and for EBoVW the best AUC and the best EER are 96.71% and 9.32%, respectively. So, more promising results for face detection were obtained by EBoVW while 4 dictionaries were used. Fig. 2 and Fig. 3 show plot of ROC for the best results achieved by both BoVW and EBoVW. Also, from the tables we see that by increasing the number of dictionaries the performance of the proposed method increases. Compared to other methods that used MIT CBCL dataset, in [16] authors achieved the best EER of 15.9%. Also authors in [17] stated that their proposed method achieved poor results on MIT CBCL dataset. In [18] authors constructed a new face dataset using MIT CBCL, AR, PIE and Yale datasets. They created this new dataset with different poses, orientations, expressions, lighting conditions, and with or without occlusions. They then rotated, cropped, and re-scaled the face images into the resolution of 24×24 pixels. They final goal was to design a general learning framework for detecting faces of various poses or under different lighting conditions. They formulated the task as a classification problem over data of multiple classes. Their approach focused on a new multi-class boosting algorithm, called MBHboost, and its integration with a cascade structure for effectively performing face detection.

The only advantage of BoVW over EBoVW is its time complexity. Due to using fuzzy c-means in EBoVW, this method is slightly slower than traditional BoVW.

4 Conclusion

This work introduces an extended version of bag of visual words model. The EBoVW uses multiple dictionary per image and also fuzzy C-mean clustering algorithm to construct histogram of words (feature vectors). In the EBoVW, more words are taken from different independent dictionaries while in the original BoVW model more words are taken from same dictionary. Thus multiple dictionary method has less storage and also more discriminative power than the original one. The experimental results on MIT CBCL face dataset proves the superiority of the proposed method in face detection problem in terms of Area Under the Curve (AUC) and Equal Error Rate (EER). The performance measures used for evaluation increases as the number of dictionary is increased for a particular value of word per dictionary.

References

1. Jafri, R., Arabnia, H.R.: A Survey of Face Detection Techniques. Journal of Information Processing Systems **5**(2) (2009)
2. Brunelli, R., Poggio, T.: Face detection: features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **15**, 1042–1052 (1993)
3. Grudin, M.A.: On internal representations in face detection systems. Pattern Detection **33**, 1161–1177 (2000)
4. Heisele, B., Wu, P.H.J., Poggio, T.: Face detection: component-based versus global approaches. Computer Vision and Image Understanding **91**, 6–21 (2003)
5. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP (2009)
6. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
7. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
8. Aly, M., Welinder, P., Munich, M., Perona, P.: Towards automated large scale discovery of image families. In: CVPR Workshop on Internet Vision (2009)
9. Aly, M.: Online learning for parameter selection in large scale image search. In: CVPR Workshop OLCV (2010)
10. Lowe, D.G.: Distinctive image features from Scale-invariant key-points. International Journal of Computer Vision **2**(60), 91–110 (2004)
11. Toews, M., Arbel, T.: Detection, Localization, and Sex Classification of Faces from Arbitrary Viewpoints and under Occlusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(9), 1567–1581 (2009)
12. Center for Biological and Computational Learning at MIT (MIT-CBCL) Face Database #1. http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html
13. Zhang, J., Marszałek, M., Lazebnik, C., Schmid, S.: Local features and kernels for classification of texture and Scene categories: a comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)
14. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 27:1–27:27 (2011)
15. Mansfield, A.J., Wayman, J.A.: Best Practices in Testing and Reporting Performance of Biometric Devices (2002)
16. Smeraldi, F.: Ranklets: orientation selective non-parametric features applied to face detection. In: Proceedings of 16th IEEE International Conference on Pattern Detection, vol. 16(3), pp. 379–382 (2002)
17. Zhu, H., Koga, T.: Face Detection based on AdaBoost Algorithm with Local Autocorrelations image. IEICE Electronics Express **7**(15), 1125–1131 (2010)
18. Lin, Y.Y., Liu, T.-L.: Robust face detection with multi-class boosting. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 680–687 (2005)

Improving Multi-adjoint Logic Programs by Unfolding Fuzzy Connective Definitions

Pedro J. Morcillo and Ginès Moreno(✉)

Department of Computing System,
University of Castilla-La Mancha, Albacete 02071, Spain
pmorcillo@dsi.uclm.es, gines.moreno@uclm.es

Abstract. Declarative programming has been classically used for solving computational problems regarding AI, knowledge representation and so on. During the last decade, Soft-Computing has emerged as a new application area specially tempting for those new generation declarative languages integrating fuzzy logic into logic programming. In many fuzzy logic programming languages, both program clauses and connective definitions admit a clear declarative, rule-based representation inspired by the well-known logic and functional programming paradigms, respectively. A powerful and promising proposal in this area is represented by the multi-adjoint logic programming approach (for which we have developed the $\mathcal{FLOPER}$ tool), where a set of (logic) PROLOG-like rules are accompanied with a set of (functional) HASKELL-like fuzzy connective definitions for manipulating truth degrees beyond the simpler case of {*true,false*}. Since these definitions can be seen as a particular case of equations and/or rewrite rules typically used in functional programming, in this paper we focus on their optimization by reusing some variants of program transformation techniques based on unfolding with a functional taste, which have been largely exploited in this last crisp (not fuzzy) setting. We also show how our method rebounds in the simplification of some computational cost measures we proposed in the past. Our approach is accompanied with some implementation and practical issues in connection with the $\mathcal{SYNTH}$ and $\mathcal{FLOPER}$ tools and the *fuzzyXPath* application we have developed in the area of the semantic web.

Keywords: Fuzzy logic programming · Connectives · Unfolding

1 Introduction

Fuzzy Logic Programming [5,7,11,15,23] is an interesting and still growing research area that agglutinates the efforts for introducing fuzzy logic into *Logic Programming* [16], in order to provide techniques and constructs for dealing with uncertainty and approximated reasoning in a natural way. Most of these systems replace the classical inference mechanism of SLD–Resolution with a fuzzy variant which

Work supported by the EU (FEDER), and the Spanish MINECO Ministry (*Ministerio de Economía y Competitividad*) under grant TIN2013-45732-C4-2-P.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 511–524, 2015.
DOI: 10.1007/978-3-319-19258-1_42

is able to handle partial truth. This is the case of *multi-adjoint logic programming* (MALP in brief, [17]), where programs are parametric to lattices modeling rich notions of truth degrees. In this framework, a program is a set of "weighted" rules together with a set of equations (*rewriting rules*) defining the repertoire of fuzzy connectives considered in a concrete lattice of truth degrees.

As we will see in Section 2, to solve a goal, i.e, a query to the system plus a substitution (initially the empty substitution, denoted by *id*), a generalization of the classical *modus ponens* inference rule called *admissible steps* are systematically applied on atoms in a similar way to classical resolution steps in pure logic programming, thus returning a state composed by a computed substitution together with an expression where all atoms have been exploited. Next, this expression is interpreted under a given lattice, hence returning a pair $\langle$*truth degree*; *substitution*$\rangle$ which is the fuzzy counterpart of the classical notion of computed answer used in pure logic programming.

As we showed in [14], this last interpretive process admits a formulation based on a transition system where each *interpretive step* solves a concrete connective on a state. In [18] we have improved such definition by explicitly expanding connective definitions and evaluating primitive (arithmetic) operators on states. The method does not alter the final set of solutions, but it has the extra ability of exhibiting the complexity of the interpretive phase in detail. Very close to this last work, in [19] we were also involved in defining abstract approaches to cost measurement, motivated by the evidence that the usual method of counting the number of derivation steps required to reach a solution was inappropriate when considering the interpretive phase. The final conclusion is that (interpretive) cost measures must rely on the "weights", that is, the number of connective calls and primitive operators evaluated directly or indirectly by the connectives appearing in each interpretive step of a given derivation.

Motivated by this fact, in this paper we are interested in reducing the complexity of connectives (also alleviating the computational cost of derivations) by safely removing all the intermediate calls performed on the equations defining the behavior of such connectives. In Section 3 we show that this process can be easily described in terms of "unfolding", a well-known, widely used, semantics-preserving program transformation rule which in most declarative paradigms is usually based on the application of computation steps on the body of program rules ([8,12–14] describe our experiences in fuzzy logic programming). The novelty of our present approach is that it is the first time that unfolding is not applied to program rules, but to connective definitions, maintaining the same final goal, i.e., generating more efficient code.

On the other hand, in our research group we have designed the $\mathcal{FLOPER}$ programming environment useful for implementing real-world applications with clear relationships regarding the soft-computing arena [1–3] (please visit `http://dectau.uclm.es/floper` and `http://dectau.uclm.es/fuzzyXPath`). Since the benefits of using the techniques described so far directly apply on the development of both tools, in Section 4 we will be concerned with some implementation issues which make use of the transformation system $\mathcal{SYNTH}$. Finally, in Section 5 we present our conclusions and provide lines of future research.

2 Multi-adjoint Logic Programming

This section summarizes the main features of multi-adjoint logic programming (see [17] for a complete formulation of this framework). We work with a first order language, $\mathcal{L}$, containing variables, constants, function symbols, predicate symbols, and several (arbitrary) connectives to increase language expressiveness: implication connectives $(\leftarrow_1, \leftarrow_2, \ldots)$; conjunctive operators (denoted by $\&_1, \&_2, \ldots$), disjunctive operators $(|_1, |_2, \ldots)$, and hybrid operators (usually denoted by $@_1, @_2$, ...), all of them are grouped under the name of "aggregators" or directly "connectives". Aggregation operators are useful to describe/specify user preferences. An aggregation operator, when interpreted as a truth function, may be an arithmetic mean, a weighted sum or in general any monotone application whose arguments are values of a complete bounded lattice L. For example, if an aggregator @ is interpreted as $[\![@]\!](x, y, z) = (3x + 2y + z)/6$, we are giving the highest preference to the first argument, then to the second, being the third argument the least significant. Although these connectives are binary operators, we usually generalize them as functions with an arbitrary number of arguments. So, we often write $@(x_1, \ldots, x_n)$ instead of $@(x_1, \ldots, @(x_{n-1}, x_n), \ldots)$. By definition, the truth function for an n-ary aggregation operator $[\![@]\!] : L^n \to L$ is required to be monotonous and fulfills $[\![@]\!](\top, \ldots, \top) = \top$, $[\![@]\!](\bot, \ldots, \bot) = \bot$.

Additionally, our language $\mathcal{L}$ contains the values of a multi-adjoint lattice, $\langle L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n \rangle$, equipped with a collection of adjoint pairs $\langle \leftarrow_i, \&_i \rangle$, where each $\&_i$ is a conjunctor which is intended to the evaluation of *modus ponens* [17]. In general, L may be the carrier of any complete bounded lattice but, for readability reasons, in the examples we shall select L as the set of real numbers in the interval $[0, 1]$. A *L-expression* is a well-formed expression composed by values and connectives of L, as well as variable symbols and *primitive operators* (i.e., arithmetic symbols such as $*, +, min$, etc...). In what follows, we assume that the truth function of any connective @ in L is given by its corresponding *connective definition*, that is, an equation or *rewriting rule* of the form $@(x_1, \ldots, x_n) = E$, where E is a L-expression not containing variable symbols apart from $x_1, \ldots, x_n$.

A *rule* is a formula $H \leftarrow_i \mathcal{B}$, where H is an atomic formula or atom (usually called the *head*) and $\mathcal{B}$ (which is called the *body*) is a formula built from atomic formulas $B_1, \ldots, B_n$ — $n \geq 0$ —, truth values of L, conjunctions, disjunctions and aggregations. A *goal* is a body submitted as a query to the system. Roughly speaking, a multi-adjoint logic program is a set of pairs $\langle \mathcal{R}; \alpha \rangle$ (we often write $\mathcal{R}$ *with* α), where $\mathcal{R}$ is a rule and α is a *truth degree* (a value of L) expressing the confidence of a programmer in the truth of the rule $\mathcal{R}$. By abuse of language, we sometimes refer a tuple $\langle \mathcal{R}; \alpha \rangle$ as a "rule".

The procedural semantics of the multi–adjoint logic language $\mathcal{L}$ can be thought as an operational phase (based on admissible steps) followed by an interpretive one. In the following, $\mathcal{C}[A]$ denotes a formula where A is a sub-expression which occurs in the –possibly empty– context $\mathcal{C}[]$. Moreover, $\mathcal{C}[A/A']$ means the replacement of A by A' in context $\mathcal{C}[]$, whereas $\mathcal{V}ar(s)$ refers to the set of

distinct variables occurring in the syntactic object s, and $\theta[\mathcal{V}ar(s)]$ denotes the substitution obtained from θ by restricting its domain to $\mathcal{V}ar(s)$.

Definition 1 (Admissible Step). *Let $\mathcal{Q}$ be a goal and let σ be a substitution. The pair $\langle \mathcal{Q}; \sigma \rangle$ is a* state *and we denote by $\mathcal{E}$ the set of states. Given a program $\mathcal{P}$, an* admissible computation *is formalized as a state transition system, whose transition relation $\overset{AS}{\rightsquigarrow} \subseteq (\mathcal{E} \times \mathcal{E})$ is the smallest relation satisfying the following* admissible rules *(where we always consider that A is the selected atom in $\mathcal{Q}$ and $mgu(E)$ denotes the* most general unifier *of an equation set E):*

1) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS}{\rightsquigarrow} \langle (\mathcal{Q}[A/v \&_i \mathcal{B}])\theta; \sigma\theta \rangle$
   if $\theta = mgu(\{A' = A\})$, $\langle A' \leftarrow_i \mathcal{B}; v \rangle$ in $\mathcal{P}$ and $\mathcal{B}$ is not empty.
2) $\langle \mathcal{Q}[A]; \sigma \rangle \overset{AS}{\rightsquigarrow} \langle (\mathcal{Q}[A/v])\theta; \sigma\theta \rangle$
   if $\theta = mgu(\{A' = A\})$ and $\langle A' \leftarrow_i ; v \rangle$ in $\mathcal{P}$.

As usual, rules are taken renamed apart. We shall use the symbols $\overset{\text{AS1}}{\rightsquigarrow}$ and $\overset{\text{AS2}}{\rightsquigarrow}$ to distinguish between computation steps performed by applying one of the specific admissible rules. Also, the application of a rule on a step will be annotated as a superscript of the $\overset{\text{AS}}{\rightsquigarrow}$ symbol.

Definition 2. *Let $\mathcal{P}$ be a program and let $\mathcal{Q}$ be a goal. An* admissible derivation *is a sequence $\langle \mathcal{Q}; id \rangle \overset{AS}{\rightsquigarrow}* \langle \mathcal{Q}'; \theta \rangle$. When $\mathcal{Q}'$ is a formula not containing atoms (i.e., a L-expression), the pair $\langle \mathcal{Q}'; \sigma \rangle$, where $\sigma = \theta[\mathcal{V}ar(\mathcal{Q})]$, is called an* admissible computed answer *(a.c.a.) for that derivation.*

Example 1. Let $\mathcal{P}$ be the following multi-adjoint logic program:

$$\begin{array}{ll} \mathcal{R}_1 : p(X) \leftarrow_{\texttt{godel}} \&_{\texttt{prod}}(|_{\texttt{luka}}(q(X), 0.6), r(X)) & \text{with } 0.9 \\ \mathcal{R}_2 : q(a) \leftarrow & \text{with } 0.8 \\ \mathcal{R}_3 : r(X) \leftarrow & \text{with } 0.7 \end{array}$$

where the labels `luka`, `godel` and `prod` mean respectively for *Łukasiewicz logic*, *Gödel intuitionistic logic* and *product logic*, that is,

$$\begin{array}{ll} |_{\texttt{luka}}(x_1, x_2) & = min(1, x_1 + x_2) \\ \&_{\texttt{prod}}(x_1, x_2) & = x_1 * x_2 \\ \leftarrow_{\texttt{godel}}(x_1, x_2) & = \ if\ (x_1 > x_2)\ then\ x_2\ \ else\ 1 \\ \&_{\texttt{godel}}(x_1, x_2) & = min(x_1, x_2) \end{array}$$

Now, we can generate the following admissible derivation (we underline the selected atoms in each step):

$$\begin{array}{ll} \langle \underline{p(X)};\ id \rangle & \overset{\text{AS1}}{\rightsquigarrow}{}^{\mathcal{R}_1} \\ \langle \&_{\texttt{godel}}(0.9, \&_{\texttt{prod}}(|_{\texttt{luka}}(\underline{q(X_1)}, 0.6), r(X_1))); \{X/X_1\} \rangle & \overset{\text{AS2}}{\rightsquigarrow}{}^{\mathcal{R}_2} \\ \langle \&_{\texttt{godel}}(0.9, \&_{\texttt{prod}}(|_{\texttt{luka}}(0.8, 0.6), \underline{r(a)})); \{X/a, X_1/a\} \rangle & \overset{\text{AS2}}{\rightsquigarrow}{}^{\mathcal{R}_3} \\ \langle \&_{\texttt{godel}}(0.9, \&_{\texttt{prod}}(|_{\texttt{luka}}(0.8, 0.6), \underline{0.7})); \{X/a, X_1/a, X_2/a\} \rangle & \end{array}$$

Here, the admissible computed answer (a.c.a.) is the pair: $\langle \&_{\texttt{godel}}(0.9, \&_{\texttt{prod}}(|_{\texttt{luka}}(0.8, 0.6), 0.7)); \{X/a\}\rangle$.

If we exploit all atoms of a goal, by applying admissible steps as much as needed during the operational phase, then it becomes a formula with no atoms (a L-expression) which can be then directly interpreted w.r.t. lattice L. Although in [17] this last process is implicitly collected in a definition similar to the previous one for describing the intended notion of fuzzy computed answer, here we prefer to model it as a new computational process (transition system) by applying the following definition we initially presented in [14] (as we will see in further sections, the cost measures proposed in this paper and previous ones [18,19], are strongly related with the behaviour and detailed definition of interpretive step):

Definition 3 (Interpretive Step). *Let $\mathcal{P}$ be a program, $\mathcal{Q}$ a goal and σ a substitution. We formalize the notion of* interpretive computation *as a state transition system, whose transition relation $\overset{IS}{\rightsquigarrow} \subseteq (\mathcal{E} \times \mathcal{E})$ is defined as the least one satisfying: $\langle Q[@(r_1, \ldots, r_n)]; \sigma\rangle \overset{IS}{\rightsquigarrow} \langle Q[@(r_1, .., r_n)/[\![@]\!](r_1, .., r_n)]; \sigma\rangle$, where $[\![@]\!]$ is the truth function of connective $@$ in the lattice $\langle L, \preceq\rangle$ associated to $\mathcal{P}$.*

Definition 4. *Let $\mathcal{P}$ be a program and $\langle Q; \sigma\rangle$ an a.c.a., that is, $\mathcal{Q}$ is a goal not containing atoms (i.e., a L-expression). An* interpretive derivation *is a sequence $\langle Q; \sigma\rangle \overset{IS}{\rightsquigarrow} * \langle Q'; \sigma\rangle$. When $Q' = r \in L$, being $\langle L, \preceq\rangle$ the lattice associated to $\mathcal{P}$, the state $\langle r; \sigma\rangle$ is called a* fuzzy computed answer *(f.c.a.) for that derivation.*

Example 2. We complete the previous derivation of Example 1 by applying 3 interpretive steps in order to obtain the final f.c.a. $\langle 0.7; \{X/a\}\rangle$, thus generating the following interpretive derivation D_1:

$$\begin{array}{ll} \langle \&_{\texttt{godel}}(0.9, \&_{\texttt{prod}}(\underline{|_{\texttt{luka}}(0.8, 0.6)}, 0.7)); \{X/a\}\rangle & \overset{\text{IS}}{\rightsquigarrow} \\ \langle \&_{\texttt{godel}}(0.9, \underline{\&_{\texttt{prod}}(1, 0.7)}); \{X/a\}\rangle & \overset{\text{IS}}{\rightsquigarrow} \\ \langle \underline{\&_{\texttt{godel}}(0.9, 0, 7)}; \{X/a\}\rangle & \overset{\text{IS}}{\rightsquigarrow} \\ \langle 0.7; \{X/a\}\rangle. & \end{array}$$

We have implemented the previous procedural principle into $\mathcal{FLOPER}$ as shown in Figure 1, where each state (containing its associated goal and substitution) is coloured in yellow and computational steps appear in blue circles (each admissible step is labeled with the used program rule, whereas label "`is`" reflects interpretive steps). Moreover, in Figure 3 we find a tree of derivations drawn again by $\mathcal{FLOPER}$, where some transitions are labeled with "`sis1`" and "`sis2`", according with the definition of the so-called *small interpretive steps* [18] that we are going to explain in the following sections.

3 Unfolding Connective Definitions

As we said in the previous section, connective definitions are equations (or *rewriting rules*) of the form $@(x_1, \ldots, x_n) = E$, where E is a L-expression which might

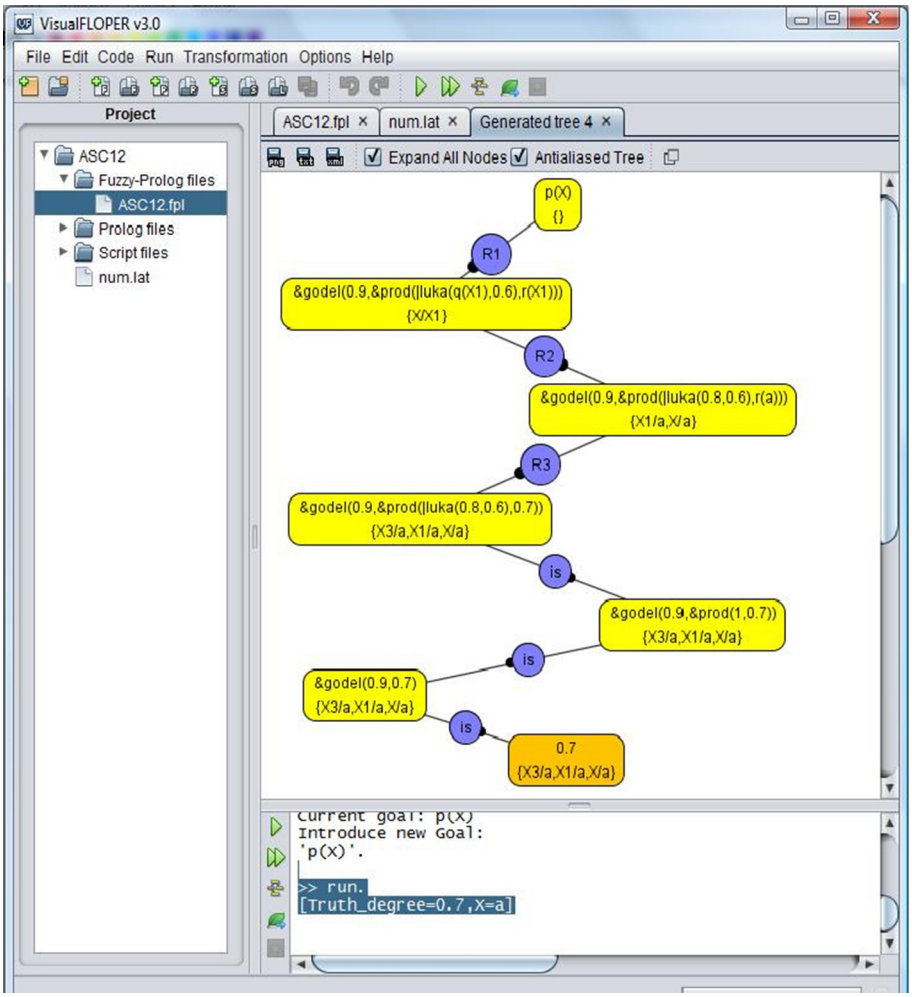


Fig. 1. The fuzzy logic programming environment $\mathcal{FLOPER}$

contain variable symbols in the set $\{x_1, \ldots, x_n\}$, as well as values, primitive operators and connectives of a multi-adjoint lattice L. The use of connectives inside the definition of other connectives is a powerful expressive resource useful not only for programmers interested in describing complex aggregators, but it also plays an important role in fuzzy transformation techniques such as the fold/unfold framework we have described in [8,12–14]. Consider for instance, the following connective definition: $@_{\texttt{complex}}(x_1, x_2) = \&_{\texttt{prod}}(|_{\texttt{luka}}(x_1, 0.6), x_2)$. This hybrid aggregator was used in [18,19] (with slight modifications) for pointing out some disturbs observed when measuring the interpretive cost associated to the execution of MALP programs.

Example 3. A simplified version of rule $\mathcal{R}_1$ in Example 1, whose body only contains an aggregator symbol is $\mathcal{R}_4 : p(X) \leftarrow_{\texttt{godel}} @_{\texttt{complex}}(q(X), r(X))$ with 0.9. Note that $\mathcal{R}_4$ has exactly the same meaning (interpretation) that $\mathcal{R}_1$, although different syntax. In fact, both of them have the same sequence of atoms in their head and bodies. The differences are regarding the set of connectives which explicitly appear in their bodies since in $\mathcal{R}_4$ we have moved $\&_{\texttt{P}}$ and $|_{\texttt{luka}}$ (as well as value 0.6) from the body of the rule (see $\mathcal{R}_1$) to the connective definition of $@_{\texttt{complex}}$.

On the other hand, a classical, simple way for estimating the computational cost required to built a derivation, consists in counting the number of computational steps performed on it. So, given a derivation D, we define:

- *operational cost* $\mathcal{O}_c(D)$, as the number of admissible steps performed in D, and
- *interpretive cost* $\mathcal{I}_c(D)$, as the number of interpretive steps done in D.

Note that the operational and interpretive costs of derivation D_1 performed in the previous section are $\mathcal{O}_c(D_1) = 3$ and $\mathcal{I}_c(D_1) = 3$, respectively. Intuitively, $\mathcal{O}_c$ informs us about the number of atoms exploited along a derivation. Similarly, $\mathcal{I}_c$ seems to estimate the number of connectives evaluated in a derivation. However, this last statement is not completely true: $\mathcal{I}_c$ only takes into account those connectives appearing in the bodies of program rules which are replicated on states of the derivation, but no those connectives recursively *nested* in the definition of other connectives, as we are going to see. Let us use rule $\mathcal{R}_4$ instead of $\mathcal{R}_1$ for generating the following derivation D_1^* which returns the same f.c.a than D_1:

$$
\begin{array}{ll}
\langle \underline{p(X)};\ id \rangle & \overset{\text{AS1}^{\mathcal{R}_4}}{\rightsquigarrow} \\
\langle \&_{\texttt{godel}}(0.9, @_{\texttt{complex}}(\underline{q(X_1)}, r(X_1)); \{X/X_1\} \rangle & \overset{\text{AS2}^{\mathcal{R}_2}}{\rightsquigarrow} \\
\langle \&_{\texttt{godel}}(0.9, @_{\texttt{complex}}(0.8, \underline{r(a)})); \{X/a, X_1/a\} \rangle & \overset{\text{AS2}^{\mathcal{R}_3}}{\rightsquigarrow} \\
\langle \&_{\texttt{godel}}(0.9, \underline{@_{\texttt{complex}}(0.8, 0.7)}); \{X/a, X_1/a, X_2/a\} \rangle & \overset{\text{IS}}{\rightsquigarrow} \\
\langle \underline{\&_{\texttt{godel}}(0.9, 0.7)}; \{X/a, X_1/a, X_2/a\} \rangle & \overset{\text{IS}}{\rightsquigarrow} \\
\langle 0.7; \{X/a, X_1/a, X_2/a\} \rangle &
\end{array}
$$

Note that, since we have exploited the same atoms with the same rules (except for the first steps performed with the equivalent rules $\mathcal{R}_1$ and $\mathcal{R}_4$, respectively) in both derivations, then $\mathcal{O}_c(D_1) = \mathcal{O}_c(D_1^*) = 3$. However, although connectives $\&_{\texttt{prod}}$ and $|_{\texttt{luka}}$ have been evaluated in both derivations, in D_1^* such evaluations have not been explicitly counted as interpretive steps, and consequently they have not been added to increase the interpretive cost measure $\mathcal{I}_c$. This unrealistic situation is reflected by the abnormal result: $\mathcal{I}_c(D_1) = 3 > 2 = \mathcal{I}_c(D_1^*)$. In [18,19] we have described two different techniques (based respectively on a redefinition of the notion of interpretive step and in the introduction of the concept of "weight of a connective") evidencing that the interpretive cost of derivation

D_1^* is not only lower, but even greater than derivation D_1. The main reason is that complex connective definitions involving calls to other aggregators consume more computational resources than other connectives which only evaluate primitive operators.

The previous example motivates the following definition, which in essence describes a technique based on classical unfolding transformations for simplifying, when possible, connective definitions by "unnesting" unnecesary calls to other connectives.

Definition 5 (C-Unfolding). *Let $\langle L, \preceq \rangle$ be a multi-adjoint lattice containing the connective definitions $@(x_1, \ldots, x_n) = E$ and $@'(x'_1, \ldots, x'_m) = E'$, such that a call to $@'$ of the form $@'(t_1, \ldots, t_m)$ appears in E. Then, the unfolding of connective $@$ w.r.t. connective $@'$ or directly, the c-unfolding of $@$, is the new equation: $@(x_1, \ldots, x_n) = E[@'(t_1, \ldots, t_m)/E'']$, where E'' is obtained from the L-expression E' by replacing each variable (formal parameter) x'_i by its corresponding value (actual parameter) t_i, $1 \leq i \leq m$, that is $E'' = E'[x'_1/t_1, \ldots, x'_m/t_m]$.*

We assume here that the rules (equations) describing connective definitions are taken renamed apart (at least one of them) before applying an unfolding step, as it is also usual with program rules in many declarative transformation tasks.

Example 4. Focusing now in the connective definition:

$$@_{\texttt{complex}}(x_1, x_2) \;=\; \&_{\texttt{prod}}(|_{\texttt{luka}}(x_1, 0.6), x_2)$$

... and remembering that $|_{\texttt{luka}}(x'_1, x'_2) = min(1, x'_1 + x'_2)$, then, we can unfold connective $@_{\texttt{complex}}$ w.r.t. connective $|_{\texttt{luka}}$ as follows:

- Firstly, we generate the "matcher" between the call $|_{\texttt{luka}}(x_1, 0.6)$ appearing in the "rhs" (right hand side) of the first equation rule and the "lhs" (left hand side) of the second rule $|_{\texttt{luka}}(x'_1, x'_2)$, thus producing links x'_1/x_1 and $x'_2/0.6$.
- Next, we apply both bindings to the rhs of the second rule, obtaining the L-expression $min(1, x_1 + 0.6)$.
- Then, this L-expression is used to replace the original call to $|_{\texttt{luka}}$ in the rhs of the first rule, producing $\&_{\texttt{prod}}(min(1, x_1 + 0.6), x_2)$.
- Finally, this last L-expression conforms the rhs of the new connective definition for $@_{\texttt{complex}}$, that is: $@_{\texttt{complex}}(x_1, x_2) = \&_{\texttt{prod}}(min(1, x_1 + 0.6), x_2)$.

Following the same method, but performing now the c-unfolding of $@_{\texttt{complex}}$ w.r.t. $\&_{\texttt{prod}}$ whose connective definition is $\&_{\texttt{prod}}(x_1, x_2) = x_1 * x_2$, we obtain the final rule defining $@_{\texttt{complex}}$ with the following shape $@_{\texttt{complex}}(x_1, x_2) = min(1, x_1 + 0.6) * x_2$. Note that the new connective definition is just a simple arithmetic expressions involving primitive operators but no calls to other connectives, as wanted. From now on, this improved definition will be referred as $@_{\texttt{unfolded}}$.

4 Implementation and Practical Issues

The ideas managed so far have been mainly inspired by our previous studies and experiences in the following two topics: fold/unfold transformations in declarative programming and cost measures in multi-adjoint logic programming ([18–20]).

Focusing on primitive functional programs, the pioneer work [6] initiates a fertile tradition in program optimization techniques based on fold/unfold transformations, which has highly attracted a wide audience in the declarative programming research community during the last three decades (see the first introduction into logic programming in [22], and then our adaptations to functional logic programming in [4,21] and fuzzy logic programming in [8,12–14]). This approach is based on the construction, by means of a *strategy* (heuristic), of a sequence of equivalent programs—called *transformation sequence* and usually denoted by $\mathcal{P}_0, \ldots, \mathcal{P}_n$ such that $\mathcal{P}_n$ is more efficient than $\mathcal{P}_0$—where each program $\mathcal{P}_i$ is obtained from the preceding ones $\mathcal{P}_0, \ldots, \mathcal{P}_{i-1}$ by using an *elementary* transformation rule. The essential rules are *folding* and *unfolding*, i.e., contraction and expansion of sub-expressions of a program using the definitions of this program (or of a preceding one).

The transformation system $\mathcal{SYNTH}$ [4,21] was initially conceived for optimizing functional logic programs (Curry, [9]), and then used for manipulating too pure functional programs (Haskell, [10]). The tool implements five basic transformation rules (unfolding, folding, abstraction, definition introduction and definition elimination) and two automatic transformation strategies (composition and tupling), in order to generate efficient sets of rewriting rules coded with Haskell/Curry syntax[1]. For the purposes of the present work, we simply need to consider the unfolding operation of the $\mathcal{SYNTH}$ system for being applied on rewriting rules modeling fuzzy connectives, as occurs with the following ones of our running example:

```
or_luka A B = min 1 (add A B)
and_prod A B = prod A B
agr_complex A B = and_prod (or_luka A (0,6)) B
```

Here, the connectives $|_{\texttt{luka}}$, $\&_{\texttt{prod}}$ and $@_{\texttt{complex}}$ are respectively denoted by `or_luka`, `and_prod` and `agr_complex`; we use the names `min`, `add` and `prod` for referring to the primitive arithmetic operators $min, +$ and $*$, respectively; and finally `A` and `B` are variable symbols. Once the previous set of connective definitions is loaded into the $\mathcal{SYNTH}$ system, it conforms the initial program in the transformation sequence denoted by "`asc 0`". Next, we must select the third rule and click on the `unfolding` button, thus obtaining the next program "`asc 1`" where the selected rule is replaced by the new fourth one `agr_complex A B = and_prod (min 1 (add A (0,6))) B`. Finally, this last rule admits a final

[1] In essence, both languages share the same syntax, but they have a different computational behaviour since Curry extends with extra logic features the pure functional dimension of Haskell.

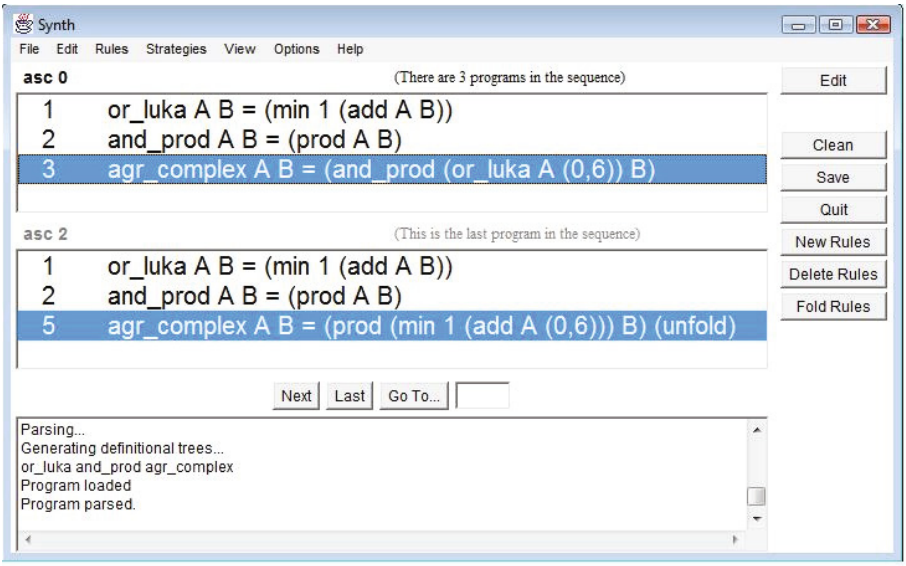


Fig. 2. The transformation system $\mathcal{SYNTH}$

unfolding step to reach our intended final definition for `agr_complex` (where no calls to other connectives appear), represented by the fifth rule `agr_complex A B = prod (min 1 (add A (0,6))) B` in program "`asc 2`". Figure 2 shows the original and final program of the transformation sequence, where the initial and improved definitions of `agr_complex` appear darked in blue.

On the other hand, our "*Fuzzy LOgic Programming Environment for Research*" $\mathcal{FLOPER}$, is able to trace the execution of goals with respect to a given MALP program and its associated lattice, by drawing derivation trees as the ones shown in Figures 1 and 3. When we choose option "`ismode=small`" then the system is able to detail the set of connective calls performed along a derivation (`sis1` steps) as well as the set of primitive operators (`sis2` steps) evaluated in the same derivation, according to the notion of *small interpretive step* formalized in [18]. Before explaining the three derivations collected on the tree drawn in Figure 3, remember that our MALP program looks like:

$$\begin{array}{ll}
\mathcal{R}_1 : p(X) \leftarrow_{\texttt{godel}} \&_{\texttt{prod}}(|_{\texttt{luka}}(q(X), 0.6), r(X)) & \text{with } 0.9 \\
\mathcal{R}_2 : q(a) \leftarrow & \text{with } 0.8 \\
\mathcal{R}_3 : r(X) \leftarrow & \text{with } 0.7 \\
\mathcal{R}_4 : p(X) \leftarrow_{\texttt{godel}} @_{\texttt{complex}}(q(X), r(X)) & \text{with } 0.9 \\
\mathcal{R}_5 : p(X) \leftarrow_{\texttt{godel}} @_{\texttt{unfolded}}(q(X), r(X)) & \text{with } 0.9
\end{array}$$

where $@_{\texttt{unfolded}}$ represents the improved definition (where no calls to other connectives are performed) of $@_{\texttt{complex}}$.

In the figure, each derivation starts with an admissible step exploiting a different MALP rule defining predicate p (i.e., $\mathcal{R}_1$, $\mathcal{R}_4$ and $\mathcal{R}_5$, respectively) and

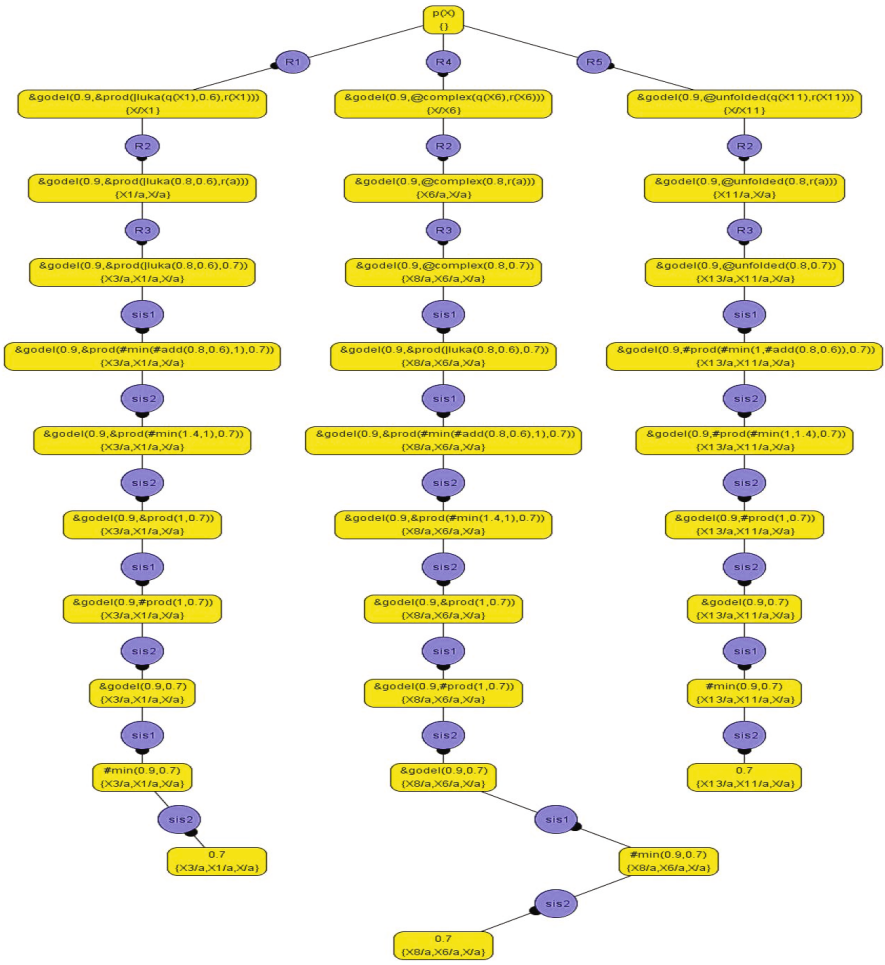


Fig. 3. Derivation tree using different definitions of fuzzy connectives

continues with rules $\mathcal{R}_2$ and $\mathcal{R}_3$ in all cases until finishing the admissible phase. From here, the interpretive phase differs in each derivation since, even when all them evaluate exactly the same set of primitive operators (so, the number of `sis2` steps do coincide), they perform a different number of direct/indirect calls to connectives (represented by `sis1` steps). So, the left-most branch in Figure 3 is shorter than the branch in the middle of the tree (as we have largely explained in the previous section), but note that the right-most branch, which makes use of the connective whose definition has been improved by means of our unfolding technique, is just the shortest one in the whole figure, as wanted.

We have just seen by means of a very simple example, that the "quality" of the connective definitions accompanying a MALP program, directly reverts on the efficiency of the applications coded with this fuzzy language. Thus, the

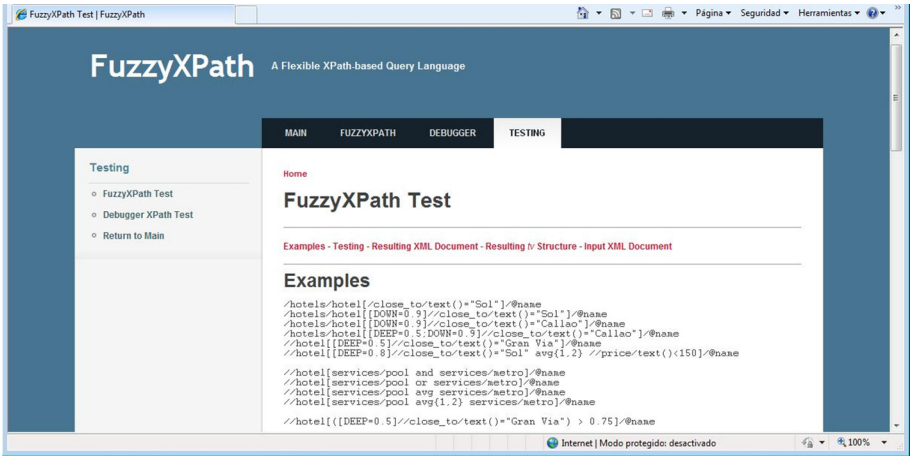


Fig. 4. An on-line work session with the `fuzzyXPath` application

unfolding technique applicable on fuzzy connectives described along this paper, is intended to play an important role in the performance of large scale programs developed with our $\mathcal{FLOPER}$ environment. To finish this section, we give some hints about the real-world *fuzzyXPath* application we have recently developed with this tool in the field of the semantic web (see Figure 4).

In [1,3] it is presented a fuzzy variant of the popular XPath query language for the flexible information retrieval on XML documents. The main purpose is to provide a repertoire of operators that offer the possibility of managing satisfaction degrees by adding structural constraints and fuzzy operators inside conditions, in order to produce a ranked sorted list of answers according to user's preferences when composing queries. For instance, if we have an XML document containing information about a set of hotels and we plan to retrieve those ones which are close to "Sol" square and have pool, an example of standard (crisp) XPath query should be: «//hotel//close_to/text="Sol" and pool]/@name». However, a more flexible query for obtaining a ranked list of hotel's names with pool and close to "Sol" as much as possible, but giving the double of importance to the feature of having pool, we could perform the following *fuzzyXPath* query, where we use two fuzzy commands (in boldface):

«//hotel[[**DEEP=0.8**]//close_to/text="Sol" **avg{1,2}** pool/@name»

Moreover, the method for debugging XPath queries presented in [2] has been implemented too with the fuzzy logic language MALP by using the $\mathcal{FLOPER}$ tool developed in our group. The method describes how XPath expressions can be manipulated for obtaining a set of alternative (graded too) formulations matching a given XML document. For each new proposed query, a "chance degree" is given representing an estimation on its deviation w.r.t. the initial expression. Our work is focused on providing to the programmer a repertoire of paths that

(s)he can use to retrieve answers. The technique can be tested on-line (see Figure 4) through `http://dectau.uclm.es/fuzzyXPath/`.

5 Conclusions and Future Work

In this paper we were concerned with the optimization of fuzzy logic connectives whose artificial, inefficient definitions could have been automatically produced by previous transformation processes applied on fuzzy MALP programs. Our technique, inspired by rewriting-based unfolding, takes profit from clear precedents largely studied and reported in other well-known declarative paradigms such as pure functional programming [6] and functional logic programming [4,21]. Anyway, such antecedents have a nature and evolution far away from our target goal for promoting the agile introduction of fuzzy logic into logic programming. We think that this work could open a fruitful research line intended to transfer a great amount of results successfully exploited by the declarative community to the new generation of fuzzy logic languages.

As many other results in our research activities around the fuzzy logic language MALP, the unfolding technique formalized and implemented in this paper for optimizing connective definitions by using the transformation system $\mathcal{SYNTH}$ [4,21], supposes a new ingredient in the development of the fuzzy logic programming environment $\mathcal{FLOPER}$ (visit `http://dectau.uclm.es/floper/`). We have just used this tool for implementing a fuzzy variant of the well-known XPath language in order to help the flexible manipulation of XML documents (see [1–3] and visit `http://dectau.uclm.es/fuzzyXPath/`). Our results reveal that the multi-adjoint logic approach serves as an excellent fuzzy logic platform for developing real world applications related with challenging research areas such as Soft-Computing, Internet and the Semantic Web.

References

1. Almendros-Jiménez, J.M., Luna, A., Moreno, G.: Fuzzy logic programming for implementing a flexible xpath-based query language. Electronic Notes in Theoretical Computer Science **282**, 3–18 (2012)
2. Almendros-Jiménez, J.M., Luna Tedesqui, A., Moreno, G.: Annotating "Fuzzy Chance Degrees" When Debugging XPath Queries. In: Rojas, I., Joya, G., Cabestany, J. (eds.) IWANN 2013, Part II. LNCS, vol. 7903, pp. 300–311. Springer, Heidelberg (2013)
3. Almendros-Jiménez, J.M., Luna, A., Moreno, G.: Fuzzy xpath through fuzzy logic programming. New Generation Computing, 42 (in press, 2015)
4. Alpuente, M., Falaschi, M., Moreno, G., Vidal, G.: Rules + Strategies for Transforming Lazy Functional Logic Programs. Theoretical Computer Science **311**(1–3), 479–525 (2004)
5. Baldwin, J.F., Martin, T.P., Pilsworth, B.W.: Fril- Fuzzy and Evidential Reasoning in Artificial Intelligence. John Wiley & Sons Inc. (1995)
6. Burstall, R.M., Darlington, J.: A Transformation System for Developing Recursive Programs. Journal of the ACM **24**(1), 44–67 (1977)

7. Guadarrama, S., Muñoz, S., Vaucheret, C.: Fuzzy Prolog: A new approach using soft constraints propagation. Fuzzy Sets and Systems **144**(1), 127–150 (2004)
8. Guerrero, J.A., Moreno, G.: Optimizing fuzzy logic programs by unfolding, aggregation and folding. Electronic Notes in Theoretical Computer Science **219**, 19–34 (2008)
9. Hanus, M. (ed.). Curry: An integrated functional logic language (vers. 0.8.3) (2012). http://www.curry-language.org
10. Hutton, G.: Programming in Haskell. Cambridge University Press (2007)
11. Ishizuka, M., Kanai, N.: Prolog-ELF incorporating fuzzy logic. In: Joshi, A.K. (eds.) Proceedings of the 9th Int. Joint Conference on Artificial Intelligence, IJCAI 1985, pp. 701–703. Morgan Kaufmann (1985)
12. Julián-Iranzo, P., Medina-Moreno, J., Morcillo, P.J., Moreno, G., Ojeda-Aciego, M.: An Unfolding-Based Preprocess for Reinforcing Thresholds in Fuzzy Tabulation. In: Rojas, I., Joya, G., Gabestany, J. (eds.) IWANN 2013, Part I. LNCS, vol. 7902, pp. 647–655. Springer, Heidelberg (2013)
13. Julián, P., Moreno, G., Penabad, J.: On Fuzzy Unfolding. A Multi-adjoint Approach. Fuzzy Sets and Systems **154**, 16–33 (2005)
14. Julián, P., Moreno, G., Penabad, J.: Operational/Interpretive Unfolding of Multi-adjoint Logic Programs. Journal of Universal Computer Science **12**(11), 1679–1699 (2006)
15. Julián, P., Moreno, G., Penabad, J., Vázquez, C.: A fuzzy logic programming environment for managing similarity and truth degrees. Electronic Proceedings in Theoretical Computer Science **173**, 71–86 (2014)
16. Lloyd, J.W.: Foundations of Logic Programming. Springer, Berlin (1987)
17. Medina, J., Ojeda-Aciego, M., Vojtáš, P.: Similarity-based Unification: a multi-adjoint approach. Fuzzy Sets and Systems **146**, 43–62 (2004)
18. Morcillo, P.J., Moreno, G.: Modeling Interpretive Steps in Fuzzy Logic Computations. In: Di Gesù, V., Pal, S.K., Petrosino, A. (eds.) WILF 2009. LNCS, vol. 5571, pp. 44–51. Springer, Heidelberg (2009)
19. Morcillo, P.J., Moreno, G.: On cost estimations for executing fuzzy logic programs. In: Arabnia, H.R., et al. (eds.) Proceedings of the 11th International Conference on Artificial Intelligence, ICAI 2009, July 13–16, Las Vegas, Nevada, USA, pp. 217–223. CSREA Press (2009)
20. Morcillo, P.J., Moreno, G., Penabad, J., Vázquez, C.: Fuzzy Computed Answers Collecting Proof Information. In: Cabestany, J., Rojas, I., Joya, G. (eds.) IWANN 2011, Part II. LNCS, vol. 6692, pp. 445–452. Springer, Heidelberg (2011)
21. Moreno, G.: A narrowing-based instantiation rule for rewriting-based fold/unfold transformations. Electronic Notes in Theoretical Computer Science **86**(3), 144–167 (2003)
22. Tamaki, H., Sato, T.: Unfold/Fold Transformations of Logic Programs. In: Tärnlund, S. (ed.) Proc. of Second Int'l Conf. on Logic Programming, pp. 127–139 (1984)
23. Vojtáš, P., Paulík, L.: Soundness and completeness of non-classical extended SLD-resolution. In Dyckhoff, R. et al. (eds.) Proc. ELP 1996 Leipzig, pp. 289–301. LNCS 1050. Springer (1996)

A Mixed Fuzzy Similarity Approach to Detect Plagiarism in Persian Texts

Hamid Ahangarbahan and Gholam Ali Montazer(✉)

School of Engineering, Tarbiat Modares University, Tehran, Iran
{h.ahangarbahan,montazer}@modares.ac.ir

Abstract. A variety of methods and metrics have been offered so far to measure the extent of similarity among various documents and plagiarism detection systems. However, most of them do not take ambiguity inherent in natural language into account. Therefore, in this paper, a new method taking lexical and semantic features and similarity measures into consideration has been proposed. In the first step, after preprocessing and removing stop word, a text was divided into two parts: general and domain-specific knowledge words. Then, the mixed lexical and semantic fuzzy inference system was designed to assess text similarity. The proposed method was evaluated on Persian paper abstracts of International Conference on e-Learning and e-Teaching (ICELET) Corpus and using IT domain knowledge ontology. The results indicated that the proposed method can achieve a rate of 79% in terms of precision and can detect 83% of the plagiarism cases.

Keywords: Plagiarism · Similarity metric · Fuzzy sets · Semantic similarity · Lexical similarity

1 Introduction

At present, the vast amount of text resources can easily be accessed on the Internet. Although such great abundance of documents provides us with fast and quick access to information, similar and duplicated data results in waste of time and confusion on the part of researchers who would like to detect the originality of a document. On the other hand, the act of using others´ ideas and words in a text document without acknowledging the sources is one of the fast growing issues in academic communities especially for the higher education institutions, which are defined as Plagiarism. This problem is categorized under Natural Language Processing (NLP) domain and in recent years, many researchers have become interested in finding out an optimal solution to detecting similar and duplicated documents. Text similarity detection can be employed in paraphrasing recognition, sentiment analyses, text clustering, etc [1].

Due to the ambiguity and subjective nature of human language, detecting text similarity and plagiarism is complex task to do. For instance, the same two texts in different content may have different meanings. Numerous studies have been conducted to detect similar documents as well as plagiarism, for example, Osman, et al., proposed semantic role labeling for sentence to detect similarity which the main drawback of

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 525–534, 2015.
DOI: 10.1007/978-3-319-19258-1_43

their methods is related to numerous computations needed to make [2]. Alzahrani and Salimi used fuzzy membership function to calculate the degree of similarity between two words. They obtained an accuracy of 54.24% on PAN09 corpus. Their algorithm suffers from time complexity and also needs to improve membership function [3]. In another study, Gupta, et al., drawing on the model offered by Alzahrani and Salimi, redefined fuzzy membership function in smaller range intervals. The authors concluded that preprocessing on part of speech (POS) level and its integration with fuzzy-based methods would contribute to effective recognition similar documents [4].

Based on their study, El-Alfy, et al., proposed a framework that employs abductive (ANN) neural networks. They used five simple and weak metrics and applied ANN algorithm on PAN 2010 but as it was the case with aforementioned research, their proposed framework suffers from time and memory complexity. Their work also does not take semantic meaning of words into consideration [5]. Zesch, et al., demonstrated that context-based measures cannot detect all forms of plagiarism and proposed to apply style-based and grammar-based measures. They came to the conclusion that for more accurate detection, all types of similarity measures should be exploited together. They also stated that text features should be properly selected so that the similarity measure works to its optimum [6].

As it is clear from the review of the literature, few researches has focused on the ambiguity of the similarity measurement and most of them have not taken the types of content and domain-specific knowledge into account. From another perspective, in these studies the methods of text similarity measurement which exert a major influence on the accuracy of evaluation are crisp. As a result, when two texts in specialized or academic domain are available for comparison, we will face uncertainty and ambiguity in detecting the correct sense of the word. Such limitation in similarity measurement reduces the effectiveness of previous methods [7-8]. To overcome these limitations, we have proposed a new method that can deal with the inadequacy of information about the similarity measure. This method deploys fuzzy linguistic variables to express experts' knowledge about text similarity and also employ lexical, semantic and stylistic features to obtain a comprehensive evaluation of text similarity and finally detects plagiarism and similar documents.

The rest of this paper is organized as follows: In the following section, plagiarism detection problem is being described. Section 3, fundamental concepts of fuzzy set theory will be represented and in Section 4, the architecture of the proposed method and numerical results are elaborated respectively. Finally, Section 5 offers the conclusions and implications of the study.

2 Problem Statement

Plagiarism detection can be defined in the simplest form as the process of locating instances of using documents or applications without citing the original reference [9] and using other materials to create documents in your own name are among the most commonly used instances of it.

Since plagiarists change the word order of sentences, transpose different sections of a text or rewrite a text by changing a word to its equivalent semantic meaning (substituting synonyms or paraphrasing), detecting these types of plagiarism is too difficult, especially when we are concerned with specific content domain texts. In addition, if the text is written in a low resource language such as Persian that has limited access to electronic resources such as WorldNet for the purpose of similarity detection, and if there are any, they are not accurate in terms of quality and comprehensiveness, we will encounter some difficulties and limitations in text similarity detection. Therefore, we need to clarify the problem statement and identify factors that affect similarities measurement in Persian to obtain a better performance. So in this paper, we consider the following question:

Given two Persian texts in specific (scientific) content domain, how can we calculate the degree of similarity between two texts with high accuracy and precision?

As it was already stated, Persian language suffers from a lack of rich resources and there are no standard datasets and corpora the same as Microsoft Research Paraphrase Corpus (MSRPC) for Persian texts; nor are there any specific content domain corpora available which pair sentences that feature the same meaning in IT domain. Due to these inadequacies, in this research we created a domain specific corpus obtained Persian paper abstracts of international conference on e-Learning and e-Teaching (PeLeT) to benchmark our method with others.

3 Fuzzy Sets Theory

Fuzzy sets theory was introduced by Lotfi Zadeh in 1965[10]. This theory is an appropriate framework for handling uncertain and imprecise data. This framework uses a set of "if-then" rules assigned to inference, in which any of these rules are defined by fuzzy sets. Fuzzy logic uses linguistic variables, which can be easily understood by humans and allows decision-making in spite of incomplete and uncertain information.

Fuzzy inference systems consist of four main components as fuzzifier, knowledge base, inference system, defuzzification. In the fuzzifier process, relationships between the inputs of system and linguistic variables are defined by fuzzy membership functions. In knowledge base component, all linguistic rules are extracted from the domain experts, that is, experts on linguistics. These rules use linguistic variables to express relationships between inputs and outputs of the system. The format of the rule is represented as[11]:

If *"the input conditions are true"* then *"the set of outputs is inference"*.

The inference system component is related to the decision part of the system and is able to infer outputs using fuzzy rules and operators. In this research, Mamdani product inference system generates outputs by using inputs based on predefined rules. And finally the defuzzification step performs the reverse of fuzzifier process and generates a crisp value of fuzzy output. There are many techniques to defuzzification. In this research, the center of gravity is exploited in defuzziness process [12].

4 The Proposed Method

In this study, a new mixed fuzzy inference system (FIS) was proposed which accomplishes the inference through two FISs, assesses similarity between two sentences and finally detects plagiarism. Fig. 1 demonstrates the conceptual structure of the proposed method. The proposed method is composed of three components: preprocessing and segmentation, features extraction and similarity measures selection and finally fuzzy inference system. These components are described in the rest of the paper.

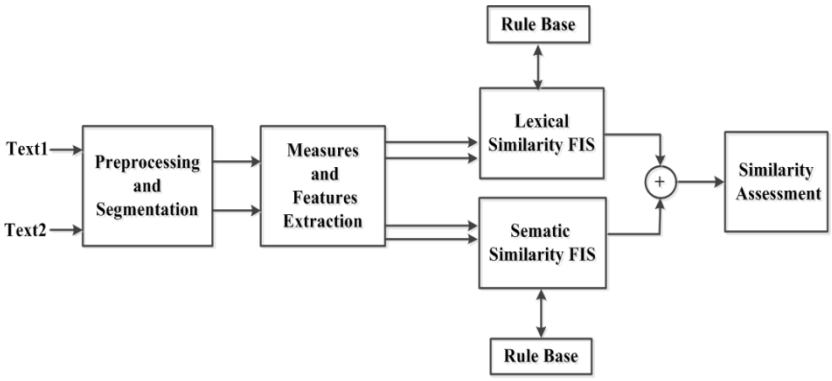


Fig. 1. The Proposed mixed FIS Method

4.1 Preprocessing and Segmentation

In this component, preprocessing operations such as stemming and stop-word removal is done in each input text. By definition, text contains one or more words that express a special meaning in the context in which it is stated. This meaning is usually expressed through domain-specific knowledge terms. Therefore, we need to disambiguate the sense of a word according to its context. To gain more precision and accuracy, in the next step each text is divided into general and domain-specific knowledge parts. In the general part, words such as 'book' and in the domain-specific knowledge part words such as 'E-Learning' are included. For instance, consider the following two sentences "I have expertise in data mining" and "John has experience in software applications and semantic web". Fig. 2 depicts the segmentation of these sentences. Domain-specific knowledge terms and words are extracted using the ontology of IT technical terms. In this research, due to the lack of such domain-specific knowledge ontology in Persian, we had to compile it.

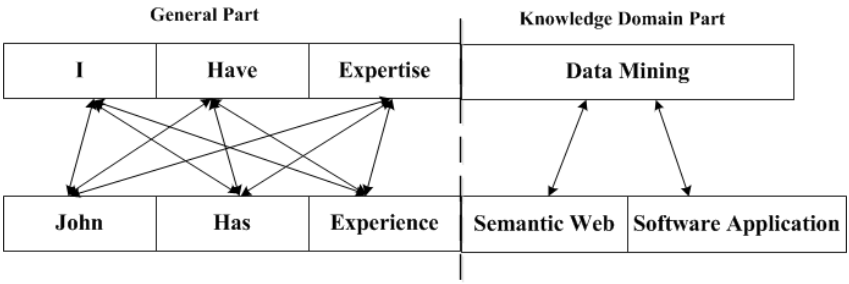


Fig. 2. Segmentation and comparing in sentence pairs

4.2 Measures and Features Extraction

After segmentation, the relevant sections are compared in a way that the general part of the first sentence is being compared to the general part of the second sentence; the same process also occurs for domain-specific knowledge parts. We used lexical- and semantic-based features and metrics to assess the extent of similarity between two the texts. These features and metrics have been categorized in two groups and are introduced in the following:

Lexical Approach Similarity. The similarity measures and features in this approach only address the surface level and do not take the meaning and senses of them into account.

Difference between the number of words in each part. The reason for choosing this feature is that the similar sentence usually varies a little bit in the number of words. Although this rule does not function as a feature to all similarity detections in general, but it serves as a supplementary feature for assessing similarity. This feature will also be exploited in semantic approach to similarity detection.

F Measure. This measure calculates the common words between the two texts by using n-grams in word or character level, divided by the length of the first and second text respectively. Then, it computes the geometric mean of the two ratios [13].

Skip-Gram Measure. This measure is similar to the n-gram but can skip between words as '*skip'* length. According to the examination on Persian IT corpus, the best skip number was found to be 3. This measure deals with word order as well as detects the common phrases in two sentences [14].

Semantic Approach to Similarity. The similarity measures and features in this approach only consider the meaning of words and phrases according to a specialized dictionary and ontology.

The Pairs of the General Words with Distance Measure. By using a dictionary, this measure calculates the value of semantic similarity between pairs of words and phrases generated by skip-gram.

The Pairs of the Domain-specific knowledge Words with Distance Measure. Using a specialized dictionary, this measure calculates the extent of semantic similarity between pairs of words and phrases. In this research, we employed Wu and Palmer's [15] edge based metric and Resink's [16] node-based metric (information content based).

Table 1 presents the complete list of measures and features used in this research. The linguistic variables of the proposed FIS were developed on the basis of these measures and features.

Table 1. Selected Features and Measures

		Term Part	
		General	**Domain Knowledge**
Similarity Approach	**Lexical**	1- Difference between the number of words 2- Skip Gram 3- F Measure	1- Difference between the number of words 2- Skip Gram 3- F Measure
	Semantic	1- The General Words Pairs Distance Measure (GWPDM)	1- The Domain knowledge edge based on Words Pairs Distance Measure (DKEWPDM) 2- The Domain knowledge information content based on Words Pairs Distance Measure (DKIWPDM)

4.3 The Mixed Text Similarity Fuzzy Inference System (MiTSiFIS)

Due to the complex nature of natural language, it is quite difficult to detect similarity between scientific and domain-specific knowledge texts. Meanwhile, the ambiguity inherent in human language prohibits us from developing efficient NLP techniques, especially where the same text might be interpreted differently with taking the context into account. By the same token, the same content might be worded differently in various paraphrases. That is why we need to gather information from experts to achieve a more precise assessment. To do so, we designed two lexical and semantic FISs that assess text similarity from a different perspective. The output of each FIS will be combined and finally the proposed mixed method can determine whether two sentences like each other. We model similarity assessment as fuzzy linguistic variables to overcome the ambiguity and vagueness of the assessment. In this research, each input variable model as trapezoidal fuzzy number. In Table 2 similarity of linguistic variable will be expressed.

Table 2. The Similarity of Linguistic Variable

Linguistic Variable	Interval
Different	(-∞,0,0.1,0.35)
Semi Similar	(0.1,0.35,0.55,0.75)
Similar	(0.65,0.75, 1,+∞)

Since assessing the similarities between the two texts is defined as a fuzzy number, inputs of FIS are also modeled as a fuzzy number. For easy and quick access, Table 3 displays all types of variables and Table 4 depicts variables used in each FIS with their correspondence.

Table 3. Lingustic Variables Types in Simialrity Assesment

Variable	Linguistic Variable	Type	Interval
Ratio of Number of Domain Knowledge to General Words (DWGWR)	Low	T1	(-∞,0,0.1,0.25)
	Middle		(0.15,0.35,0.55,0.8)
	High		(0. 5,0.75, 1,+∞)
Similarity Assess-ment between word pairs	Different	T2	(-∞,0,0.1,0.35)
	Semi Similar		(0.1,0.35,0.55,0.75)
	Similar		(0.65,0.75, 1,+∞)

Table 4. FISs Variables

FIS	Part	Variable	Variable Type	Variable Name
Lexical Similarity	General	Skip-Gram	T1	V1
		F Measure	T1	V2
	Knowledge Domain	Skip-Gram	T1	V3
		F Measure	T1	V4
	All Sentence	DWGWR	T2	V5
Semantic Similarity	General	GWPDM	T1	V6
	Knowledge Domain	DKEWPDM	T1	V7
		DKIWPDM	T1	V8
	All Sentence	DWGWR	T2	V9

4.4 Fuzzy Rules Base

After precisely defining FIS variables, system rules were deduced and developed by conducting an interview with a group of five natural language and domain experts. Table 5 and 6 demonstrates some fuzzy rules for lexical and semantic FIS respectively. In this rule base, we consider fuzzy “AND” for t-norm and operators among fuzzy

variables and the operator for rules aggregation is s-norm. For example, rule 5 in Table 5 is as follow:

If "*F Measure in General Part is Low*", AND "*F Measure in Knowledge domain Part is Low*", THEN "*Similarity is Different*".

Table 5. Some rules of FIS lexical similarity

Rule #	**Variable Name**					**Similarity**
	V1	**V2**	**V3**	**V4**	**V5**	
1	-	High	-	Low	Low	Similar
2	-	High	Low	Low	Middle	Semi Simialr
3	-	High	Middle	Low	Middle	Semi Simialr
4	Low	Middle	Low	Low	Middle	Different
5	-	Low	-	Low	-	Different

Table 6. Some rules of FIS semantic similarity

Rule #	Variable Name				Similarity
	V6	V7	V8	V9	
1	Low	Low	Low		Different
2	Low	Low	Middle	Low	Different
3	Low	Middle	High	Low	Semi Similarr
4	-	-	-	High	Similarr
5	High	Low	Low	Low	Similarr

4.5 Numerical Results

In this section, the numerical results of the proposed method will be reported. The proposed method was implemented in Visual Studio Environment and used MATLAB fuzzy toolbox to simulate FISs. Since the objective of the research was to detect plagiarism in Persian scientific texts and as mentioned there was not such a corpus, we had to create PeLeT corpus in E-Learning domain knowledge. This corpus contains 810 sentence pairs equally assigned to "*Different*", "*Semi Similar*" and "Similar" class. The first sentence of pairs gathered from papers' abstracts presented in ICELET and a group of expert domain generated a paired sentence in one of the randomly assigned class. The average length of sentence consisted of 13 words. After preprocessing and segmentation, the proposed method applied to the corpus. To build PeLeT corpus, we carefully followed suggestions prescribed in [17] that enable us to state general conclusions.Table 7 indicates the confusion matrix of the proposed method.

Table 7. Confusion matrix of proposed method

		Predicted Class		
		Similar	**Semi Similar**	**Different**
Actual Class	**Similar**	213	32	25
	Semi Similar	28	209	32
	Different	217	36	23

To evaluate the performance of the proposed method, we used recall, precision and F-measure that are widely employed in text mining. We also intended to cast more light on the comparison of the efficiency of the proposed algorithm and given the fact that the proposed method has been applied to PeLeT Corpus, we experimented the three methods mentioned in the literature with PeLeT Corpus to assess the efficiency of our method. These methods implemented as same mentioned in the literature. Table 8 illustrates the results of the implementation of the proposed method and compares it with the three methods. The results show that proposed method outperforms the other methods. In future work, we plan to implement other machine learning methods such as support vector machines and artificial neural network.

Table 8. The result performance

Method	F Measure	Precision	Recall
The Proposed Method	**0.81**	0.795	0.83
Gupta, et al.[4]	0.67	0.66	0.69
Alzahrani and Salimi[3]	0.63	0.61	0.65
Mihalcea and Corley[18]	0.59	0.57	0.62

5 Conclusion

In this paper, a mixed fuzzy inference system method was proposed to overcome the ambiguity and elevate the degree of confidence in the similarity measurement in Persian texts. In this method, in the first step, after preprocessing and stop word removal, the text is divided into general and domain-specific knowledge parts; then, appropriate features are extracted and similarity metrics are calculated. Two fuzzy inference systems were designed and the base of rules is extracted from the group of experts and finally the outputs of two FISs are integrated. With regard to the fact that the proposed method was applied to PeLeT Corpus, we also carried out tests using the three methods proposed in the literature to measure it efficiency. Such a comparison was thought to throw light on the efficiency of the proposed algorithm. The results show that the proposed method outperforms than others and gained accuracy rate of 81% which increases precision and recall measure.

References

1. Wang, Y., Hodges, J.: Document clustering with semantic analysis. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences. HICSS 2006, vol. 3. IEEE (2006)
2. Osman, A.H., Salim, N., Binwahlan, M.S., Alteeb, R., Abuobieda, A.: An improved plagiarism detection scheme based on semantic role labeling. Applied Soft Computing **12**(5), 1493–1502 (2012)
3. Alzahrani, S., Salim, N.: Fuzzy semantic-based string similarity for extrinsic plagiarism detection. Braschler and Harman (2010)
4. Gupta, R., et al.: UoW: NLP techniques developed at the University of Wolverhampton for Semantic Similarity and Textual Entailment. SemEval **2014**, 785 (2014)
5. El-Alfy, E.-S.M., et al.: Boosting paraphrase detection through textual similarity metrics with abductive networks. Applied Soft Computing **26**, 444–453 (2015)
6. Zesch, D.B.T., Gurevych, I.: Text reuse detection using a composition of text similarity measures. In: Proceedings of COLING, vol. 1, pp. 167–184 (2012)
7. Barrón-Cedeño, A., Rosso, P.: On automatic plagiarism detection based on *n*-grams comparison. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 696–700. Springer, Heidelberg (2009)
8. Kent, C.K., Salim, N.: Features based text similarity detection. arXiv preprint arXiv:1001.3487 (2010)
9. Joy, M., Luck, M.: Plagiarism in programming assignments. IEEE Transactions on Education **42**(2), 129–133 (1999)
10. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Springer, US (1974)
11. Huang, Y.-P., et al.: An intelligent approach to detecting the bad credit card accounts. In: Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications. ACTA Press (2007)
12. Rutkowski, L., Cpalka, K.: Flexible neuro-fuzzy systems. IEEE Transactions on Neural Networks **14**(3), 554–574 (2003)
13. Metzler, D., et al.: Similarity measures for tracking information flow. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management. ACM (2005)
14. Mikolov, T., et al.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
15. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics (1994)
16. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
17. Rus, V.R.B., Lintean, M.: On paraphrase identification corpora. In: Proceeding on the International Conference on Language Resources and Evaluation (LREC 2014) (2014)
18. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. AAAI. **6** (2006)

A Neural-Network-Based Robust Observer for Simultaneous Unknown Input Decoupling and Fault Estimation

Piotr Witczak(✉), Marcin Mrugalski, Krzysztof Patan, and Marcin Witczak

Institute of Control and Computation Engineering,
University of Zielona Góra, ul. Podgórna 50, 65–246 Zielona Góra, Poland
{P.Witczak,M.Mrugalski,K.Patan,M.Witczak}@issi.uz.zgora.pl

Abstract. The paper deals with the problem of neural-network based on robust unknown input observer design for the fault diagnosis. Authors review the recent development in the area of robust observers for non-linear discrete-time systems and propose less restrictive procedure for design of the $\mathcal{H}_\infty$ observer. The approach guaranties simultaneously the unknown input decoupling and the fault estimation. The paper presents an unknown input observer design that reduces to a set of linear matrix inequalities. The final part of the paper presents an illustrative example devoted to fault diagnosis of the wind turbine.

Keywords: Takagi-Sugeno systems · System identification · Artificial neural networks · Sector non-linearities · Observer · Robustness · Fault diagnosis

1 Introduction

Over the years the problem of the fault diagnosis has awaited for several efficient solutions for wide class of linear systems. Among them the parity relation, observers and parameter identification methods can be distinguished [7,15]. Such methods are usually efficient in both fault detection and fault estimation [17] tasks. Essentially the information about the level of the fault is important because it is required for the efficient Fault-Tolerant Control systems [4,16,18]. Unfortunately, in the case of Fault Diagnosis (FD) of non-linear systems [9,12] and especially in the fault estimation tasks such methods are still not sufficient. In order to solve such problem the neural networks can be applied. The idea of their application for fault estimation task seems to be very attractive especially that they are efficient in the non-linear dynamic system identification. In order to achieve this goal the state-space description of the neural model is required. Such description enables to spin off such a model to the form of the Takagi-Sugeno (T-S) fuzzy model [10]. The properties of such a representation

The work was financed as a research project with the science funds with the kind support of the National Science Centre in Poland.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 535–548, 2015.
DOI: 10.1007/978-3-319-19258-1_44

allow to apply the advanced algorithms like the Unknown Input Observer (UIO) for simultaneous state and fault estimation of non-linear discrete-time systems. Authors show in the paper that the fault estimation can be also perceived as the estimation of an unknown input. It should be underlined that such strategy has received considerable attention in the literature [13,15,20]. The paper revisits recent results in this area [3], and provides less restrictive design procedure for the design of a robust UIO. The general UIO strategy and the $\mathcal{H}_\infty$ framework are used to design a robust fault estimation scheme. The main advantage of the proposed approach boils down to its simplicity, i.e., it reduces to solving a set of Linear Matrix inequalities (LMIs), which can be efficiently solved with modern computational packages. Therefore for completeness of the paper authors present a novel non-linear system identification and fault isolation scheme, which boils down to five steps:

- Train neural network to mimic non-linear dynamic system;
- Obtain weights of the neurons;
- Find the set of linear models for T-S fuzzy representation;
- Find the multi-variable membership function;
- Design the robust UIO for the fault estimation.

The paper is organized as follows. Section 2 presents the state space neural model which can be applied in the non-linear dynamic system identification. In section 3 the methodology of the neural model description by the T-S fuzzy model is developed. Subsequently, the section provides a novel UIO design procedure for the fault estimation purposes, which relaxes restrictive conditions of the competitive approaches. Section 4 delivers an illustrative example showing the effectiveness of the proposed neural model based robust UIO in the fault estimation tasks. Finally, section 5 is devoted to the concluding remarks.

2 State Space Neural Network

Let us consider a discreet-time non-linear system

$$\boldsymbol{x}_{k+1} = g(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{1}$$

where $\boldsymbol{x}_k \in \Re^n$ and $\boldsymbol{u}_k \in \Re^r$ are the state vector and system input, respectively. Function $g : \Re^n \times \Re^r \to \Re^n$ is an unknown continous mapping of the states and inputs. It should be underlined that the system (1) can be modeled by a Multi Layer Perceptron (MLP) ([8]) consisting of large enough but finite number p of neurons included in a single hidden layer. In such case the MLP can be described by the following relation:

$$f(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{u}}_k) = \boldsymbol{W}_0 \sigma(\boldsymbol{W}_x \tilde{\boldsymbol{x}}_k + \boldsymbol{W}_u \tilde{\boldsymbol{u}}_k + \tilde{\boldsymbol{W}}_b) + \epsilon_x, \tag{2}$$

where $\boldsymbol{W}_0 \in \Re^{n \times p}$, $\boldsymbol{W}_x \in \Re^{p \times n}$ and $\boldsymbol{W}_u \in \Re^{p \times r}$ are the output and hidden layer weights respectively, $\sigma(\bullet)$ is any activation (odd) function, e.g. hyperbolic tangent, however in [15] it is proven that the choice of $\sigma(\bullet)$ is irrelevant. Let

us assume that due to a training process ϵ, representing training error, is small enough to be ignored.

$$\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{W}_0\sigma(\boldsymbol{W}_x\tilde{\boldsymbol{x}}_k + \boldsymbol{W}_u\tilde{\boldsymbol{u}}_k + \tilde{\boldsymbol{W}}_b). \tag{3}$$

Rewriting (3) [1] we can shift the operating point to the origin removing bias from $\sigma(\bullet)$. Let us assume there exist an equilibrium point $(\tilde{\boldsymbol{x}}^o, \tilde{\boldsymbol{u}}^o)$ such as

$$\tilde{\boldsymbol{x}}^o = \boldsymbol{W}_0\sigma(\boldsymbol{W}_x\tilde{\boldsymbol{x}}^o + \boldsymbol{W}_u\tilde{\boldsymbol{u}}^o + \tilde{\boldsymbol{W}}_b). \tag{4}$$

Let us define new activation function:

$$\begin{aligned}\sigma_n(\boldsymbol{W}_x\boldsymbol{x}_k + \boldsymbol{W}_u\boldsymbol{u}_k) &= \sigma(\boldsymbol{W}_x\boldsymbol{x}_k + \boldsymbol{W}_x\tilde{\boldsymbol{x}}^o + \boldsymbol{W}_u\boldsymbol{u}_k \\ &+ \boldsymbol{W}_u\tilde{\boldsymbol{u}}^o + \tilde{\boldsymbol{W}}_b) - \sigma(\boldsymbol{W}_x\tilde{\boldsymbol{x}}^o + \boldsymbol{W}_u\tilde{\boldsymbol{u}}^o + \tilde{\boldsymbol{W}}_b) \\ &= \sigma(\boldsymbol{W}_x(\boldsymbol{x}_k + \tilde{\boldsymbol{x}}^o) + \boldsymbol{W}_u(\boldsymbol{u}_k + \tilde{\boldsymbol{u}}^o) + \tilde{\boldsymbol{W}}_b) - \sigma(\boldsymbol{W}_b)\end{aligned} \tag{5}$$

taking into according a new bias $\boldsymbol{W}_b = \boldsymbol{W}_x\tilde{\boldsymbol{x}}^o + \boldsymbol{W}_u\tilde{\boldsymbol{u}}^o + \tilde{\boldsymbol{W}}_b$ and shifted states $\boldsymbol{x}_k = \tilde{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}^o$ and inputs $\boldsymbol{u}_k = \tilde{\boldsymbol{u}}_k - \tilde{\boldsymbol{u}}^o$. Hence adding and substracting factor $\boldsymbol{W}_0\sigma(\boldsymbol{W}_b)$ to (3) and looking for $\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{x}}_{k+1} - \tilde{\boldsymbol{x}}^o$ gives

$$\begin{aligned}\boldsymbol{x}_{k+1} + \tilde{\boldsymbol{x}}^o =& \sigma(\boldsymbol{W}_x(\boldsymbol{x}_k + \tilde{\boldsymbol{x}}^o) + \boldsymbol{W}_u(\boldsymbol{u}_k + \tilde{\boldsymbol{u}}^o) + \tilde{\boldsymbol{W}}_b) \\ &+ \boldsymbol{W}_0\sigma(\boldsymbol{W}_b) - \boldsymbol{W}_0\sigma(\boldsymbol{W}_b) \\ =& \sigma_n(\boldsymbol{W}_x\boldsymbol{x}_k + \boldsymbol{W}_u\boldsymbol{u}_k) + \boldsymbol{W}_0\sigma(\boldsymbol{W}_b),\end{aligned} \tag{6}$$

and assuming that $\tilde{\boldsymbol{x}}^o = \boldsymbol{W}_0\sigma(\boldsymbol{W}_b)$ relation (6) receives the following form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{W}_0\sigma_n(\boldsymbol{W}_x\boldsymbol{x}_k + \boldsymbol{W}_u\boldsymbol{u}_k). \tag{7}$$

3 T-S Fuzzy Representation for System Identification

This section provides the description of system boundedness due to the sector non-linearities of neurons' activation functions. Authors proposes a set-based description of T-S matrices. This will ease later calculus and provide clearer explanation of identification procedure. Looking closer to (7) it can be checked, that the i-th state is represented as follows:

$$x_{k+1}^i = \sum_{j=1}^{p} w_0^{(ij)}\sigma_n\left(\sum_{a=1}^{n} w_x^{(ja)}x_k^{(a)} + \sum_{b=1}^{r} w_u^{(jb)}u_k^{(b)}\right), \tag{8}$$

then knowing that all inputs to the $\sigma_n(\bullet)$ are bounded, we can estimate the effective range of neural network input arguments as

$$\xi_{min}^{(j)} = \inf_{\forall k}\left\{\sum_{a=1}^{n} w_x^{(ja)}x_k^{(a)} + \sum_{b=1}^{r} w_u^{(jb)}u_k^{(b)}\right\}, \tag{9}$$

$$\xi_{max}^{(j)} = \sup_{\forall k}\left\{\sum_{a=1}^{n} w_x^{(ja)}x_k^{(a)} + \sum_{b=1}^{r} w_u^{(jb)}u_k^{(b)}\right\}, \tag{10}$$

where index j represents the neuron number. Therefore the sector bounds for $\xi_{min}^{(j)} \leq \xi^{(j)} \leq \xi_{max}^{(j)}$ can be defined in the form:

$$l^{(j)} = \inf_{\forall \xi^{(j)}} \left\{ \frac{\sigma_n(\xi^{(j)})}{\xi^{(j)}} \right\} \quad \text{and} \quad r^{(j)} = \sup_{\forall \xi^{(j)}} \left\{ \frac{\sigma_n(\xi^{(j)})}{\xi^{(j)}} \right\}. \tag{11}$$

The bounds (11) can equvalently be written as:

$$l^{(j)} \xi^{(j)2} \leq \sigma_n(\xi^{(j)}) \xi^{(j)} \leq r^{(j)} \xi^{(j)2}. \tag{12}$$

Remark 1. For hyperbolic tangent function with normalized inputs, all $l^{(j)}$ and $r^{(j)}$ are identical, moreover, $r^{(j)}$ is independent form the range of the input arguments. For normalized input range, e.g $[-1, 1]$ such values are $l^{(j)} = 0.7616$, $r^{(j)} = 0.9992$, for all $j = 1 \dots l$.

Therefore there exist vertices

$$\begin{aligned} \boldsymbol{W}_l &= \left\{ w_l^{(ij)} \in \boldsymbol{W}_l : w_l^{(ij)} = w_0^{(ij)} l^{(j)}; i = 1 \dots n, j = 1 \dots p \right\}, \\ \boldsymbol{W}_r &= \left\{ w_r^{(ij)} \in \boldsymbol{W}_r : w_r^{(ij)} = w_0^{(ij)} r^{(j)}; i = 1 \dots n, j = 1 \dots p \right\}, \end{aligned} \tag{13}$$

which belongs to set

$$\mathbb{W} = \left\{ \boldsymbol{W}_k : w_k^{(ij)} \in \boldsymbol{W}_k, w_k^{(ij)} = b w_r^{(ij)} + (b-1) w_l^{(ij)}; \forall_k = 1 \dots 2^p \right\}, \tag{14}$$

where $b \in \{0, 1\}$ and $b_{(10)} = k_{(2)}(j)$ and $w_0^{(ij)} \in \boldsymbol{W}_0$.

To synthesize the T-S fuzzy model form the (14) two further steps are required. Find a proper activation function, multiplying unique elements of (14) by $\boldsymbol{W}_x$ and $\boldsymbol{W}_u$ respectively. Thus a desired T-S discrete system can be expressed as

$$\boldsymbol{x}_{k+1} = \sum_{i=1}^{2^p} h_i(\boldsymbol{s}_k) [\boldsymbol{A}^i \boldsymbol{x}_k + \boldsymbol{B}^i \boldsymbol{u}_k] = \boldsymbol{A}(h_k) \boldsymbol{x}_k + \boldsymbol{B}(h_k) \boldsymbol{u}_k, \tag{15}$$

$h_i(\boldsymbol{s}_k)$ needs to be defined as

$$h_i(\boldsymbol{s}_k) = \prod_{j=1}^{p} (b h_r^{(j)}(\boldsymbol{s}_{k,i}) + (b-1) h_l^{(j)}(\boldsymbol{s}_{k,i})), \tag{16}$$

where $b \in \{0, 1\}$ and $b_{(10)} = i_{(2)}(j)$ and $\sum_i^{2^p} h_i(\boldsymbol{s}_k) = 1$ and

$$h_r^{(j)}(\xi) = \frac{\sigma_n(\xi^{(j)}) - l^{(j)} \xi^{(j)}}{(r^{(j)} - l^{(j)}) \xi^{(j)}}, \quad h_l^{(j)}(\xi) = \frac{r^{(j)} \xi^{(j)} - \sigma_n(\xi^{(j)})}{(r^{(j)} - l^{(j)}) \xi^{(j)}}, \tag{17}$$

which is an extension of (11). Thus making the T-S identification procedure complete.

Let us consider the T-S system form (15) extended by an unknown input and faults along with disturbances, in form

$$\boldsymbol{x}_{k+1} = \sum_{i=1}^{M} h_i(\boldsymbol{s}_k)[\boldsymbol{A}^i\boldsymbol{x}_k + \boldsymbol{B}^i\boldsymbol{u}_k + \boldsymbol{B}^i\boldsymbol{f}_k + \boldsymbol{D}^i\boldsymbol{d}_k + \boldsymbol{W}_1^i\boldsymbol{w}_k], \tag{18}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{W}_2\boldsymbol{w}_k, \tag{19}$$

that alternatively can be expressed as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}(h_k)\boldsymbol{x}_k + \boldsymbol{B}(h_k)\boldsymbol{u}_k + \boldsymbol{B}(h_k)\boldsymbol{f}_k + \boldsymbol{D}(h_k)\boldsymbol{d}_k + \boldsymbol{W}_1(h_k)\boldsymbol{w}_k, \tag{20}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{W}_2\boldsymbol{w}_k. \tag{21}$$

Therefore to achieve an unknown input decoupling effect along with fault estimation, a novel UIO scheme is proposed:

$$\boldsymbol{z}_{k+1} = \sum_{i=1}^{M} h_i(\boldsymbol{s}_k)[\boldsymbol{N}^i\boldsymbol{z}_k + \boldsymbol{G}^i\boldsymbol{u}_k + \boldsymbol{L}^i\boldsymbol{y}_k + \boldsymbol{T}\boldsymbol{B}^i\hat{\boldsymbol{f}}_k], \tag{22}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{z}_k - \boldsymbol{E}\boldsymbol{y}_k, \tag{23}$$

$$\hat{\boldsymbol{f}}_{k+1} = \hat{\boldsymbol{f}}_k + \boldsymbol{F}(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k), \tag{24}$$

or

$$\boldsymbol{z}_{k+1} = \boldsymbol{N}(h_k)\boldsymbol{z}_k + \boldsymbol{G}(h_k)\boldsymbol{u}_k + \boldsymbol{L}(h_k)\boldsymbol{y}_k + \boldsymbol{T}\boldsymbol{B}(h_k)\hat{\boldsymbol{f}}_k, \tag{25}$$

$$\hat{\boldsymbol{x}}_k = \boldsymbol{z}_k - \boldsymbol{E}\boldsymbol{y}_k, \tag{26}$$

$$\hat{\boldsymbol{f}}_{k+1} = \hat{\boldsymbol{f}}_k + \boldsymbol{F}(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k), \tag{27}$$

where $\boldsymbol{f}_k \in \Re^r$ and $\boldsymbol{d}_k \in \Re^q$ stands for fault and unknown input respectively. Since the UIO structure is given, it is possible to provide its design procedure. As it was postulated, while the effect of $\boldsymbol{w}_k$ will not be eliminated, but its influence on the performance of the UIO will be minimized. For that, let us assume that [19]:

$$l_2 = \left\{\mathbf{w} \in \mathbb{R}^n \middle|\ \|\mathbf{w}\|_{l_2} < +\infty\right\}, \tag{28}$$

$$\|\mathbf{w}\|_{l_2} = \left(\sum_{k=0}^{\infty} \|\boldsymbol{w}_k\|^2\right)^{\frac{1}{2}}. \tag{29}$$

Therefore subsequent objective is to design the observer (22)–(23) in such a way that the state estimation error $\boldsymbol{e}_k = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k$ is asymptotically convergent and the following upper bound is guaranteed:

$$\|\boldsymbol{e}\|_{l_2} \leq \omega\|\mathbf{w}\|_{l_2}, \tag{30}$$

where $\omega > 0$ is a prescribed disturbance attenuation level. Using (23) and (19), it can be shown that the state estimation receives form:

$$
\begin{aligned}
\boldsymbol{e}_{k+1} &= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k)\Bigg(\boldsymbol{T}\left[\boldsymbol{A}^i\boldsymbol{x}_k + \boldsymbol{B}^i\boldsymbol{u}_k + \boldsymbol{B}^i\boldsymbol{f}_k - \boldsymbol{B}^i\hat{\boldsymbol{f}}_k + \boldsymbol{D}^i\boldsymbol{d}_k + \boldsymbol{W}_1^i\boldsymbol{w}_k\right] \\
&\quad - \boldsymbol{N}^i\boldsymbol{z}_k - \boldsymbol{G}^i\boldsymbol{u}_k - \boldsymbol{L}^i\boldsymbol{y}_k + \boldsymbol{E}\boldsymbol{W}_2\boldsymbol{w}_{k+1}\Bigg) \\
&= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k)\Bigg(\left[\boldsymbol{T}\boldsymbol{A}^i - \boldsymbol{N}^i\boldsymbol{T} - \boldsymbol{L}^i\boldsymbol{C}\right]\boldsymbol{x}_k + \left[\boldsymbol{T}\boldsymbol{B}^i - \boldsymbol{G}^i\right]\boldsymbol{u}_k \\
&\quad + \boldsymbol{T}\boldsymbol{D}^i\boldsymbol{d}_k + \boldsymbol{T}\boldsymbol{B}^i\left[\boldsymbol{f}_k - \hat{\boldsymbol{f}}_k\right] + \left[\boldsymbol{T}\boldsymbol{W}_1^i - \boldsymbol{L}^i\boldsymbol{W}_2\right]\boldsymbol{w}_k \\
&\quad + \boldsymbol{N}^i\boldsymbol{e}_k + \boldsymbol{E}\boldsymbol{W}_2\boldsymbol{w}_{k+1}\Bigg),
\end{aligned}
\tag{31}
$$

where $\boldsymbol{T} = \boldsymbol{I} + \boldsymbol{E}\boldsymbol{C}$, while its evolution is described by:

$$
\begin{aligned}
\boldsymbol{e}_{k+1} &= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k)\left[\boldsymbol{N}^i\boldsymbol{e}_k + \boldsymbol{T}\boldsymbol{B}^i\boldsymbol{e}_{f,k} + \boldsymbol{T}\boldsymbol{W}_1^i\boldsymbol{w}_k - \boldsymbol{L}^i\boldsymbol{W}_2\boldsymbol{w}_k\right] \\
&\quad + \boldsymbol{E}\boldsymbol{W}_2\boldsymbol{w}_{k+1} = \boldsymbol{N}(h_k)\boldsymbol{e}_k + \boldsymbol{T}\boldsymbol{B}(h_k)\boldsymbol{e}_{f,k} + \boldsymbol{T}\boldsymbol{W}_1(h_k)\boldsymbol{w}_k \\
&\quad - \boldsymbol{L}(h_k)\boldsymbol{W}_2\boldsymbol{w}_k + \boldsymbol{E}\boldsymbol{W}_2\boldsymbol{w}_{k+1},
\end{aligned}
\tag{32}
$$

To eliminate the state from the state estimation error it is necessary to set

$$
\boldsymbol{N}^i\boldsymbol{T} = \boldsymbol{T}\boldsymbol{A}^i - \boldsymbol{L}^i\boldsymbol{C} \tag{33}
$$

The effect of the input on the estimation error is eliminated by setting a design parameter:

$$
\boldsymbol{G}^i = \boldsymbol{T}\boldsymbol{B}^i, \quad i = 1, \ldots, M. \tag{34}
$$

Finally the decouple the unknown inputs one of the the design parameters is set:

$$
\boldsymbol{T}\boldsymbol{D}^i = \boldsymbol{0}. \tag{35}
$$

Thus, equation (31) boils down to

$$
\begin{aligned}
\boldsymbol{e}_{k+1} &= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k)\left[\boldsymbol{N}^i\boldsymbol{e}_k + \boldsymbol{T}\boldsymbol{B}^i\boldsymbol{e}_{f,k} + \boldsymbol{T}\boldsymbol{W}_1^i\boldsymbol{w}_k - \boldsymbol{L}^i\boldsymbol{W}_2\boldsymbol{w}_k\right] \\
&\quad + \boldsymbol{E}\boldsymbol{W}_2\boldsymbol{w}_{k+1} = \boldsymbol{N}(h_k)\boldsymbol{e}_k + \boldsymbol{T}\boldsymbol{B}(h_k)\boldsymbol{e}_{f,k} + \boldsymbol{T}\boldsymbol{W}_1(h_k)\boldsymbol{w}_k \\
&\quad - \boldsymbol{L}^i\boldsymbol{W}_2\boldsymbol{w}_k + \boldsymbol{E}\boldsymbol{W}_2\boldsymbol{w}_{k+1}.
\end{aligned}
\tag{36}
$$

A the same time the fault estimation error

$$
\boldsymbol{e}_{f,k} = \boldsymbol{f}_k - \hat{\boldsymbol{f}}_k, \tag{37}
$$

can be expressed as:

$$\begin{aligned} \boldsymbol{e}_{f,k+1} &= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k) \left[\boldsymbol{f}_{k+1} - \hat{\boldsymbol{f}}_{k+1} \right] \\ &= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k) \left[\boldsymbol{f}_{k+1} - \hat{\boldsymbol{f}}_k - \boldsymbol{FCe}_k - \boldsymbol{FW}_2 \boldsymbol{w}_k \right] \\ &= \sum_{i=1}^{M} h_i(\boldsymbol{s}_k) \left[\boldsymbol{f}_{k+1} - \boldsymbol{f}_k + \boldsymbol{e}_{f,k} - \boldsymbol{FCe}_k - \boldsymbol{FW}_2 \boldsymbol{w}_k \right] \\ &= \boldsymbol{f}_{k+1} - \boldsymbol{f}_k + \boldsymbol{e}_{f,k} - \boldsymbol{FCe}_k - \boldsymbol{FW}_2 \boldsymbol{w}_k. \end{aligned} \tag{38}$$

Therefore to couple both fault and state estimate errors the supervector of $\bar{\boldsymbol{e}}_{f,k}$ can be created as follows:

$$\bar{\boldsymbol{e}}_{f,k+1} = \begin{bmatrix} \boldsymbol{e}_{k+1} \\ \boldsymbol{e}_{f,k+1} \end{bmatrix}, \tag{39}$$

we obtain

$$\bar{\boldsymbol{e}}_{k+1} = \begin{bmatrix} \boldsymbol{N}(h_k) & \boldsymbol{TB}(h_k) \\ -\boldsymbol{FC} & \boldsymbol{I} \end{bmatrix} \bar{\boldsymbol{e}}_k + \begin{bmatrix} \boldsymbol{TW}_1(h_k) - \boldsymbol{L}(h_k)\boldsymbol{C} & \boldsymbol{0} & \boldsymbol{EW}_2 \\ -\boldsymbol{FW}_2 & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \bar{\boldsymbol{w}}_k, \tag{40}$$

where:

$$\begin{aligned} \boldsymbol{\varepsilon}_k = \boldsymbol{f}_{k+1} - \boldsymbol{f}_k, \quad \bar{\boldsymbol{w}}_k &= \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\varepsilon}_k \\ \boldsymbol{w}_{k+1} \end{bmatrix}, \\ \hat{\boldsymbol{f}}_{k+1} &= \hat{\boldsymbol{f}}_k + \boldsymbol{F}\left[\boldsymbol{Cx}_k + \boldsymbol{W}_2 \boldsymbol{w}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_k\right] \\ &= \hat{\boldsymbol{f}}_k + \boldsymbol{FCe}_k + \boldsymbol{FW}_2 \boldsymbol{w}_k. \end{aligned} \tag{41}$$

For further deliberations it is assumed that $\boldsymbol{\varepsilon}_k \in l_2$. To simplify(40) it is assumed

$$\bar{\boldsymbol{e}}_{k+1} = \bar{\boldsymbol{A}}(h_k)\bar{\boldsymbol{e}}_k + \bar{\boldsymbol{W}}(h_k)\bar{\boldsymbol{w}}_k, \tag{42}$$

where:

$$\bar{\boldsymbol{A}}(h_k) = \begin{bmatrix} \boldsymbol{N}(h_k) & \boldsymbol{TB}(h_k) \\ -\boldsymbol{FC} & \boldsymbol{I} \end{bmatrix}, \quad \bar{\boldsymbol{W}}(h_k) = \begin{bmatrix} \boldsymbol{TW}_1(h_k) - \boldsymbol{L}(h_k)\boldsymbol{C} & \boldsymbol{0} & \boldsymbol{EW}_2 \\ -\boldsymbol{FW}_2 & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix}. \tag{43}$$

Theorem 1. *For a prescribed disturbance attenuation level $\omega > 0$ ($\omega = \sqrt{2}\mu$) for the state and fault estimation error (39), the $\mathcal{H}_\infty$ observer design problem for the system (18)–(19) and the observer (22)–(23) is solvable if there exist matrices $\boldsymbol{P}^i \succ 0$, $\boldsymbol{M}^i$, $\boldsymbol{X}^i$,$\boldsymbol{Q}$ and $\boldsymbol{U}$, $(i = 1, \ldots, M)$ such that the following constraints are satisfied:*

$$\sum_{i=1}^{M} h_i(s_k) \sum_{j=1}^{M} h_j(s_{k+1}) \Theta_j^i \prec 0, \tag{44}$$

with

$$\Theta_j^i = \begin{bmatrix} -\boldsymbol{P}_1^i + \boldsymbol{I} & -\boldsymbol{P}_2^{iT} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^{iT} & \boldsymbol{C}^T\boldsymbol{Q}^T \\ -\boldsymbol{P}_2^i & -\boldsymbol{P}_3^i + \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}^{iT}\boldsymbol{T}^T\boldsymbol{U}_1^T & \boldsymbol{U}_2^T \\ \boldsymbol{0} & \boldsymbol{0} & -\mu^2\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{W}_1^{iT}\boldsymbol{T}^T\boldsymbol{U}_1^T - \boldsymbol{W}_2^{iT}\boldsymbol{X}^{iT} & \boldsymbol{W}_2^T\boldsymbol{Q}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\mu^2\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{U}_2^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\mu^2\boldsymbol{I} & \boldsymbol{W}_2^T\boldsymbol{E}^T\boldsymbol{U}_2^T & \boldsymbol{0} \\ \boldsymbol{M}^i & \boldsymbol{U}_1\boldsymbol{T}\boldsymbol{B}^i & \boldsymbol{U}_1\boldsymbol{T}\boldsymbol{W}_1^i - \boldsymbol{X}^i\boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{U}_2\boldsymbol{E}\boldsymbol{W}_2 & \boldsymbol{P}_1^j - \boldsymbol{U}_1 - \boldsymbol{U}_1^T & \boldsymbol{P}_2^{jT} \\ \boldsymbol{Q}\boldsymbol{C} & \boldsymbol{U}_2 & \boldsymbol{Q}\boldsymbol{W}_2 & \boldsymbol{U}_2 & \boldsymbol{0} & \boldsymbol{P}_2^j & \boldsymbol{P}_3^j - \boldsymbol{U}_2 - \boldsymbol{U}_2^T \end{bmatrix} \tag{45}$$

where:

$$\boldsymbol{Q} = -\boldsymbol{U}_2\boldsymbol{F}, \quad \boldsymbol{M}^i = \boldsymbol{U}_1\boldsymbol{N}^i, \quad \boldsymbol{X}^i = \boldsymbol{U}_1\boldsymbol{L}^i,$$
$$\boldsymbol{P}^i = \begin{bmatrix} \boldsymbol{P}_1^i & \boldsymbol{P}_2^{iT} \\ \boldsymbol{P}_2^i & \boldsymbol{P}_3^i \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_2 \end{bmatrix}$$

and

$$\boldsymbol{M}^i\boldsymbol{T} = \boldsymbol{U}_1\boldsymbol{T}\boldsymbol{A}^i - \boldsymbol{X}^i$$

Proof. The problem of $\mathcal{H}_\infty$ observer design [19] is to determine the gain matrices $\boldsymbol{L}^i$ such that:

$$\lim_{k\to\infty} \bar{\boldsymbol{e}}_0 = \boldsymbol{0} \quad \text{for} \quad \bar{\boldsymbol{w}}_k = \boldsymbol{0}, \tag{46}$$

$$\|\bar{\boldsymbol{e}}\|_{l_2} \leq \omega\|\bar{\boldsymbol{w}}\|_{l_2} \quad \text{for} \quad \bar{\boldsymbol{w}}_k \neq \boldsymbol{0},\ \bar{\boldsymbol{e}}_0 = \boldsymbol{0}. \tag{47}$$

In this paper, it is proposed that it is sufficient to find a Lyapunov function V_k for $k = 0, \ldots \infty$ such that:

$$\Delta V + \bar{\boldsymbol{e}}_h^T\bar{\boldsymbol{e}}_k - \mu^2\bar{\boldsymbol{w}}_k^T\bar{\boldsymbol{w}}_k < 0, \tag{48}$$

where $\Delta V_k = V_{k+1} - V_k$, $\mu > 0$. Indeed, if $\boldsymbol{w}_k = \boldsymbol{0}$, for $k = 0, \ldots, \infty$ then (48) boils down to:

$$\Delta V_k + \bar{\boldsymbol{e}}_k^T\bar{\boldsymbol{e}}_k < 0,\ k = 0, \ldots \infty, \tag{49}$$

and hence $\Delta V_k < 0$, which leads to (46). If $\boldsymbol{w}_k \neq \boldsymbol{0}$ for $k = 0, \ldots, \infty$, then (48) yields

$$J = \sum_{k=0}^{\infty}(\Delta V_k + \bar{\boldsymbol{e}}_k^T\bar{\boldsymbol{e}}_k - \mu^2\bar{\boldsymbol{w}}_k^T) < 0, \tag{50}$$

which can be written as

$$J = -V_0 + \sum_{k=0}^{\infty}\bar{\boldsymbol{e}}_k^T\bar{\boldsymbol{e}}_k - \mu^2\sum_{k=0}^{\infty}\bar{\boldsymbol{w}}_k^T\bar{\boldsymbol{w}}_k < 0. \tag{51}$$

Knowing that $V_0 = 0$ for $\boldsymbol{e}_0 = 0$, (51) leads to (47) with $\omega = \sqrt{2}\mu$. Since the general framework for designing the robust observer is given, then the following form of the Lyapunov function is proposed:

$$V_k = \bar{\boldsymbol{e}}_k^T\boldsymbol{P}(h_k)\bar{\boldsymbol{e}}_k. \tag{52}$$

Therefore

$$\Delta V = \bar{\boldsymbol{e}}_{k+1}^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{e}}_{k+1} - \bar{\boldsymbol{e}}_k^T \boldsymbol{P}(h_k)\bar{\boldsymbol{e}}_k, \tag{53}$$

where

$$\Delta V + \bar{\boldsymbol{e}}_k^T \bar{\boldsymbol{e}}_k - \mu^2 \bar{\boldsymbol{w}}_k^T \bar{\boldsymbol{w}}_k < 0, \tag{54}$$

which leads to

$$\begin{aligned}
&\left[\bar{\boldsymbol{A}}(h_k)\bar{\boldsymbol{e}}_k + \bar{\boldsymbol{W}}(h_k)\bar{\boldsymbol{w}}_k\right]^T \boldsymbol{P}(h_{k+1}) \left[\bar{\boldsymbol{A}}(h_k)\bar{\boldsymbol{e}}_k + \bar{\boldsymbol{W}}(h_k)\bar{\boldsymbol{w}}_k\right] \\
&- \bar{\boldsymbol{e}}_k^T \boldsymbol{P}(h_k)\bar{\boldsymbol{e}}_k + \bar{\boldsymbol{e}}_h^T \bar{\boldsymbol{e}}_k - \mu^2 \bar{\boldsymbol{w}}_k^T \bar{\boldsymbol{w}}_k = \\
&\bar{\boldsymbol{e}}_k^T \left[\bar{\boldsymbol{A}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{A}}(h_k) - \boldsymbol{P}(h_k) + \boldsymbol{I}\right] \bar{\boldsymbol{e}}_k \\
&+ \bar{\boldsymbol{w}}_k^T \left[\bar{\boldsymbol{W}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{W}}(h_k)\right] \bar{\boldsymbol{w}}_k \\
&+ \bar{\boldsymbol{e}}_k^T \left[\bar{\boldsymbol{A}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{W}}(h_k)\right] \bar{\boldsymbol{w}}_k \\
&+ \bar{\boldsymbol{w}}_k^T \left[\bar{\boldsymbol{A}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{A}}(h_k)\right] \bar{\boldsymbol{e}}_k - \mu^2 \bar{\boldsymbol{w}}_k^T \bar{\boldsymbol{w}}_k < 0.
\end{aligned} \tag{55}$$

Thus, by defining $\boldsymbol{v}_k = [\bar{\boldsymbol{e}}_k^T, \bar{\boldsymbol{w}}_k^T]^T$, it can be shown that

$$\boldsymbol{v}_k^T \begin{bmatrix} \bar{\boldsymbol{A}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{A}}(h_k) - \boldsymbol{P}(h_k) + \boldsymbol{I} & \bar{\boldsymbol{A}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{W}}(h_k) \\ \bar{\boldsymbol{W}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{W}}(h_k) & \bar{\boldsymbol{W}}(h_k)^T \boldsymbol{P}(h_{k+1})\bar{\boldsymbol{W}}(h_k) \end{bmatrix} \boldsymbol{v}_k < 0, \tag{56}$$

then

$$\begin{bmatrix} \bar{\boldsymbol{A}}(h_k)^T \\ \bar{\boldsymbol{W}}(h_k)^T \end{bmatrix} \boldsymbol{P}(h_{k+1}) \left[\bar{\boldsymbol{A}}(h_k)\ \bar{\boldsymbol{W}}(h_k)\right] + \begin{bmatrix} -\boldsymbol{P}(h_k) + \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\mu^2 \boldsymbol{I} \end{bmatrix} \prec 0. \tag{57}$$

Applying Lemma 1 from [5,18] to (57) leads to

$$\begin{bmatrix} -\boldsymbol{P}(h_k) + \boldsymbol{I} & \boldsymbol{0} & \bar{\boldsymbol{A}}(h_k)^T \boldsymbol{U}^T \\ \boldsymbol{0} & -\mu^2 \boldsymbol{I} & \bar{\boldsymbol{W}}(h_k)^T \boldsymbol{U}^T \\ \boldsymbol{U}\bar{\boldsymbol{A}}(h_k) & \boldsymbol{U}\bar{\boldsymbol{W}}(h_k) & \boldsymbol{P}(h_{k+1}) - \boldsymbol{U} - \boldsymbol{U}^T \end{bmatrix} \prec 0, \tag{58}$$

which by substituting

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_2 \end{bmatrix}, \tag{59}$$

implies

$$\boldsymbol{U}\bar{\boldsymbol{A}}(h_k) = \begin{bmatrix} \boldsymbol{U}_1 \boldsymbol{N}(h_k) & \boldsymbol{U}_1 \boldsymbol{T} \boldsymbol{B}(h_k) \\ -\boldsymbol{U}_2 \boldsymbol{F} \boldsymbol{C} & \boldsymbol{U}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}(h_k) & \boldsymbol{U}_1 \boldsymbol{T} \boldsymbol{B}(h_k) \\ -\boldsymbol{Q} \boldsymbol{C} & \boldsymbol{U}_2 \end{bmatrix}, \tag{60}$$

$$\begin{aligned}
\boldsymbol{U}\bar{\boldsymbol{W}}(h_k) &= \begin{bmatrix} \boldsymbol{U}_1 \boldsymbol{T} \boldsymbol{W}_1(h_k) - \boldsymbol{U}_1 \boldsymbol{L}(h_k) \boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{U}_2 \boldsymbol{E} \boldsymbol{W}_2 \\ -\boldsymbol{U}_2 \boldsymbol{F} \boldsymbol{W}_2 & \boldsymbol{U}_2 & \boldsymbol{0} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{U}_1 \boldsymbol{T} \boldsymbol{W}_1(h_k) - \boldsymbol{X}(h_k) \boldsymbol{W}_2 & \boldsymbol{0} & \boldsymbol{U}_2 \boldsymbol{E} \boldsymbol{W}_2 \\ -\boldsymbol{Q} \boldsymbol{W}_2 & \boldsymbol{U}_2 & \boldsymbol{0} \end{bmatrix},
\end{aligned} \tag{61}$$

with

$$\boldsymbol{Q} = -\boldsymbol{U}_2\boldsymbol{F}, \quad M(h_k) = \boldsymbol{U}_1\boldsymbol{N}(h_k), \quad \boldsymbol{X}(h_k) = \boldsymbol{U}_1\boldsymbol{L}(h_k) \tag{62}$$

thus leading (58) to the following LMI

$$\begin{bmatrix} -\boldsymbol{P}_1(h_k)+\boldsymbol{I} & -\boldsymbol{P}_2(h_k)^T & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{P}_2(h_k) & -\boldsymbol{P}_3(h_k)+\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\mu^2\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\mu^2\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}(h_k) & \boldsymbol{U}_1\boldsymbol{T}\boldsymbol{B}(h_k) & \boldsymbol{U}_1\boldsymbol{T}\boldsymbol{W}_1(h_k)-\boldsymbol{X}(h_k)\boldsymbol{W}_2(h_k) & \boldsymbol{0} \\ \boldsymbol{Q}\boldsymbol{C} & \boldsymbol{U}_2 & \boldsymbol{Q}\boldsymbol{W}_2 & \boldsymbol{U}_2 \end{bmatrix}$$
$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{M}(h_k)^T & \boldsymbol{C}^T\boldsymbol{Q}^T \\ \boldsymbol{0} & \boldsymbol{B}(h_k)^T\boldsymbol{T}^T\boldsymbol{U}_1^T & \boldsymbol{U}_2^T \\ \boldsymbol{0} & \boldsymbol{W}_1(h_k)^T\boldsymbol{T}^T\boldsymbol{U}_1^T-\boldsymbol{W}_2^T\boldsymbol{X}(h_k)^T & \boldsymbol{W}_2^T\boldsymbol{Q}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{U}_2^T \\ -\mu^2\boldsymbol{I} & \boldsymbol{W}_2^T\boldsymbol{E}^T\boldsymbol{U}_2^T & \boldsymbol{0} \\ \boldsymbol{U}_2\boldsymbol{E}\boldsymbol{W}_2 & \boldsymbol{P}(h_{k+1})-\boldsymbol{U}_1-\boldsymbol{U}_1^T & \boldsymbol{P}_2(h_{k+1})^T \\ \boldsymbol{0} & \boldsymbol{P}_2(h_{k+1}) & \boldsymbol{P}_3(h_{k+1})-\boldsymbol{U}_2-\boldsymbol{U}_2^T \end{bmatrix} \prec \boldsymbol{0}. \tag{63}$$

which leads to (44) and (45) and completes the proof.
Note that (44) requires further relaxation procedure in order to be tractable within the effective LMI framework. A basic sufficient solution to this problem were described in [14] and further improved by many researchers (see, e.g., [6] and the references therein). As indicated in [6], the conditions provided by [11] lead to a good compromise between complexity and conservatism, which in the case (44) leads to the following lemma:

Lemma 1. *Condition* (44) *is fulfilled providing the following conditions hold:*

$$\Theta_j^i \prec \boldsymbol{0}, \quad i \in \{1,\ldots,M\}, \tag{64}$$

$$\frac{2}{M-1}\Theta_j^i + \Theta_j^i \prec \boldsymbol{0}, \quad i,j \in \{1,\ldots,M\}, \quad i \neq j. \tag{65}$$

Finally, the design procedure boils down to obtaining $\boldsymbol{E}$ from (35) (bearing in mind that $\boldsymbol{T} = \boldsymbol{I} + \boldsymbol{E}\boldsymbol{C}$), determining (34), solving (64)–(65) with respect to $\boldsymbol{P}^i \succ 0, \boldsymbol{M}^i, \boldsymbol{Q}, \boldsymbol{X}^i, \boldsymbol{U}$ $(i = 1,\ldots,M)$ as well as calculating: $\boldsymbol{F} = -\boldsymbol{U}_2^{-1}\boldsymbol{Q}$, $\boldsymbol{N}^i = \boldsymbol{U}_1^{-1}\boldsymbol{M}^i$ and $\boldsymbol{L}^i = \boldsymbol{U}_1^{-1}\boldsymbol{X}^i$.

4 Illustrative Example

In order to show the effectiveness of the proposed approach in the fault estimation tasks let us consider a three blade variable-speed, variable-pitch wind turbine [2]. To derive a T-S fuzzy model, the non-linear functions of pitch angle

and wind speed were used to obtain the input and output data, which was subsequently used to train a MLP consisting of $p = 4$ neurons in hidden layer, according to the procedure described in Sect. 3. Therefore, a set of 16 state matrices $\boldsymbol{A}^i$ and respective input matrices $\boldsymbol{B}^i$ were obtained. The membership function was constructed using (16). It should be underlined that the standard Levenberg-Marquadt method for training of the MLP was applied.

Figure 1 illustrates decreasing of the quality error during the training of the MLP. As it can be seen, training error decays at decent rate, and ends with a reasonable level.

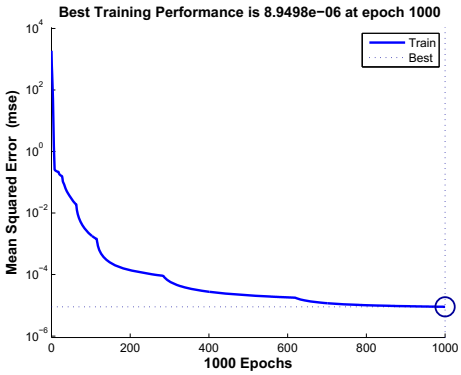


Fig. 1. The evolution of MSE of MLP

The quality of estimation of the states x_2 and x_4 are presented in Figs. 2 and 3. Such states represent the torque angle of the shaft and the pitch angle of the blades. From the above results it can be seen that the designed observer converges rapidly to the state x_2 and x_4 and tracks it with a high precision.

The obtained during the system identification T-S fuzzy model of the diagnosed system was used to design the UIO for the fault estimation purpose accord-

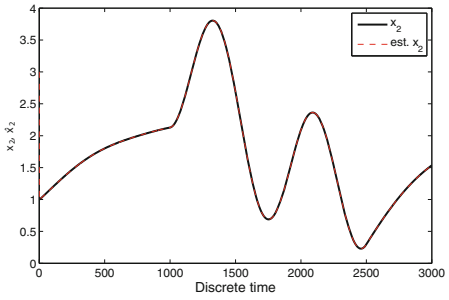


Fig. 2. State x_2 and its estimate

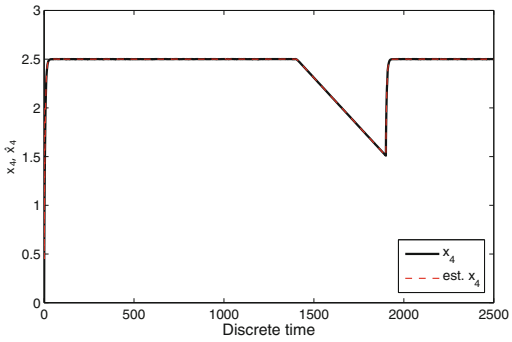


Fig. 3. State x_4 and its estimate

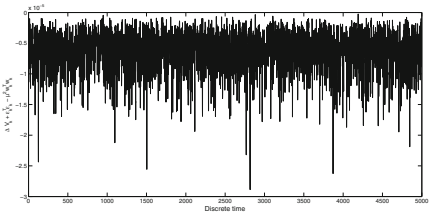


Fig. 4. Evolution of $\Delta V_k + \bar{\boldsymbol{e}}_k^T \bar{\boldsymbol{e}}_k - \mu^2 \bar{\boldsymbol{w}}_k^T \bar{\boldsymbol{w}}_k$

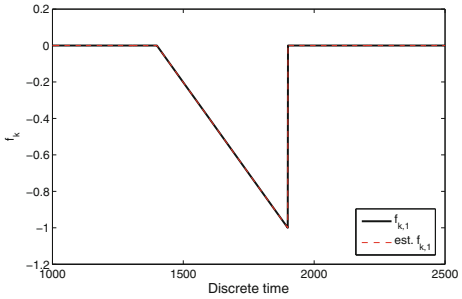


Fig. 5. Fault $f_{k,1}$ and its estimate

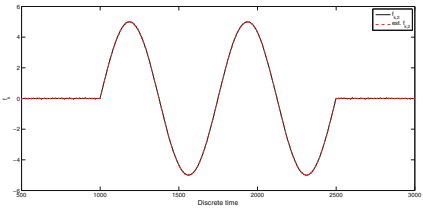


Fig. 6. Fault $f_{k,2}$ and its estimate

ing to the procedure described in Sect. 3. Figure 4 shows the evolution of Lyapunov function $\Delta V_k + \bar{\boldsymbol{e}}_k^T \bar{\boldsymbol{e}}_k - \mu^2 \bar{\boldsymbol{w}}_k^T \bar{\boldsymbol{w}}_k$ proving the effectiveness of the UIO performance. To show the effectiveness of the developed approach in the fault estimation approach, the following fault scenario was used:

$$\boldsymbol{f}_{k,1} = \begin{cases} -\frac{k-1400}{1900} & 1400 \leq k \leq 1900, \\ 0 & \text{otherwise} \end{cases}$$

$$\boldsymbol{f}_{k,2} = \begin{cases} \sin(l\pi) & 1000 \leq k \leq 2500 \wedge l = (k-1750)/375 \\ 0 & \text{otherwise} \end{cases}$$

Figure 5 depicts the estimate of the first fault concerning the pitch angle, e.g. pressure loss. The second part of the fault - the generator shaft misalignment is shown in Fig. 6. As it can be seen both faults are tracked with good performance and are estimated properly.

5 Concluding Remarks

In the paper a new methodology of estimation of faults for the non-linear discrete-time systems on the basis of the MLP was proposed. In particular, the T-S fuzzy representation of the state-space neural model is developed. Such representation allows to propose a new design procedure of a robust UIO. The strategy presented in the paper can be used to design the less restrictive fault identification scheme guaranteeing a prescribed disturbance attenuation level is achieved with respect to the state estimation error and a convergence of the observer itself. The effectiveness of the proposed approach in the tasks of the faults estimation was presented on the example of variable-speed, variable-pitch wind turbine.

References

1. Bendtsen, J.D., Trangbk, K.: Robust quasi-lpv control based on neural state space models. IEEE Transactions on Neural Networks, 355–368 (2002)
2. Bououden, S., Chadli, M., Filali, S., El Hajjaji, A.: Fuzzy model based multivariable predictive control of a variable speed wind turbine: {LMI} approach. Renewable Energy **37**(1), 434–439 (2012)
3. Chadli, M., Karimi, H.R.: Robust observer design for unknown inputs takagi-sugeno models. IEEE Transactions on Fuzzy Systems **21**(1), 158–164 (2013)
4. De Oca, S., Puig, V., Witczak, M., Dziekan, L.: Fault-tolerant control strategy for actuator faults using lpv techniques: application to a two degree of freedom helicopter. International Journal of Applied Mathematics and Computer Science **22**(1), 161–171 (2012)
5. de Oliveira, M.C., Bernussou, J., Geromel, J.C.: A new discrete-time robust stability condition. Systems and Control Letters **37**(4), 261–265 (1999)
6. Guerra, T.M., Kruszewski, A., Lauber, J.: Discrete Tagaki-Sugeno models for control: Where are we? Annual Reviews in Control **33**(1), 37–47 (2009)
7. Iserman, R.: Fault Diagnosis Applications: Model Based Condition Monitoring, Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems. Springer-Verlag, Berlin (2011)

8. Lu, S., Basar, T.: Robust nonlinear system identification using neural-network models. IEEE Transactions on Neural Networks **9**(3), 407–429 (1998)
9. Mrugalski, M.: An unscented kalman filter in designing dynamic gmdh neural networks for robust fault detection. International Journal of Applied Mathematics and Computer Science **23**(1), 157–169 (2013)
10. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modelling and control. IEEE Trans. Systems, Man and Cybernetics **15**(1), 116–132 (1985)
11. Tuan, H.D., Apkarian, P., Narikiyo, T., Yamamoto, Y.: Parameterized linear matrix inequality techniques in fuzzy control system design. IEEE Transactions on Fuzzy Systems **9**(2), 324–332 (2001)
12. Varga, A., Ossmann, D.: {LPV} model-based robust diagnosis of flight actuator faults. Control Engineering Practice **31**, 135–147 (2014)
13. Veluvolu, K.C., Kim, M.Y., Lee, D.: Nonlinear sliding mode high-gain observers for fault estimation. International Journal of Systems Science **42**(7), 1065–1074 (2011)
14. Wang, H.O., Tanaka, K., Griffin, M.F.: An approach to fuzzy control of nonlinear systems: stability and design issues. IEEE Transactions on Fuzzy Systems **4**(1), 14–23 (1996)
15. Witczak, M.: Fault Diagnosis and Fault-Tolerant Control Strategies for Non-Linear Systems. Lectures Notes in Electrical Engineering, vol. 266. Springer International Publisher, Heidelberg (2014)
16. Witczak, M., Witczak, P.: Efficient predictive fault-tolerant control for non-linear systems. In: Korbicz, J., Kowal, M. (eds.) Intelligent Systems in Technical and Medical Diagnostics. AISC, vol. 230, pp. 65–76. Springer, Heidelberg (2013)
17. Witczak, M., Witczak, P., Korbicz, J., Aubrun, C.: Robust and efficient predictive ftc: application to wind turbines. In: Conference on Control and Fault-Tolerant Systems - SysTol 2013, Nice, France, pp. 371–376 (2013)
18. Witczak, P. Luzar, M., Witczak, M., Korbicz, J.: A robust fault-tolerant model predictive control for linear parameter-varying systems. In: Proceedings of Methods and Models in Automation and Robotics - MMAR, pp. 462–467 (2014)
19. Zemouche, A., Boutayeb, M., Bara, G.I.: Observer for a class of Lipschitz systems with extension to $\mathcal{H}_\infty$ performance analysis. Systems and Control Letters **57**(1), 18–27 (2008)
20. Polycarpou, M.M., Prisini, T., Zhang, X.: Fault diagnosis of a class of nonlinear uncertain systems with lipschitz nonlinearities using adaptive estimation. Automatica **46**(2), 290–299 (2010)

Consequences of Structural Differences Between Hierarchical Systems While Fuzzy Inference

Begum Mutlu[1(✉)], Ebru A. Sezer[2], and Hakan A. Nefeslioglu[3]

[1] Department of Computer Engineering, Gazi University, Ankara, Turkey
begummutlu@gazi.edu.tr
[2] Department of Computer Engineering, Hacettepe University, Ankara, Turkey
ebruakcapinarsezer@gmail.com
[3] Technical Sciences Vocational School, Hacettepe University, Ankara, Turkey
hanefeslioglu@hacettepe.edu.tr

Abstract. Hierarchical fuzzy systems are proposed to handle the *curse of dimensionality* problem sourced from the use of single fuzzy inference systems with a large number of input parameters. While they are being used in various research problems, each of them is based on a constant hierarchic structure. In this study, this strategy is criticized because it is argued that using a constant hierarchic structure does not guarantee to obtain the most accurate solution for the problem. To observe the effects of structural differences on the prediction performance, experiments are performed on two logical gates by not only utilizing different structures but also different defuzzifiers. In the findings of the experiments, it is proved that the structural variations directly affect the systems' output, and this differentiation cannot be overcome by changing the defuzzifiers. In addition none of the utilized structures can provide the outputs of equivalent single system. It can be concluded that while applying the hierarchical fuzzy systems on any problem, different structures should be considered to find out the most accurate one that can be constantly utilized for that problem.

Keywords: Hierarchical fuzzy systems · Mamdani style fuzzy inference · Hierarchic structures

1 Introduction

Hierarchical fuzzy systems originate from the interoperation of hierarchically linked fuzzy subsystems. In these systems each individual subsystem makes a prediction based on its own inputs and rule set, and then provides data to its successive layer to be used as an input in successor's rule base. The construction of these hierarchical fuzzy systems is encouraged by two different factors. The first one is the necessity of problem combination. In this kind of hierarchies each subsystem is modeled to solve one independent problem, and the outputs of them are used to make another prediction in order to bring a solution to more complex one. In other words, these hierarchies are used to combine sub-problems

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 549–560, 2015.
DOI: 10.1007/978-3-319-19258-1_45

which are modeled by individual subsystems, and how the hierarchical system should be designed is inarguably depend on the expert. The second one, which is essentially investigated here, is the bottleneck of *curse of dimensionality* [1]. This type of dimensionality problem basically occurs due to the increase in number of input parameters. The increase, in question both complicates the fuzzy rule creation logically and rises the computational cost of inference, since it causes an exponential growth in the number of fuzzy rules [2].

It is obvious that the *curse of dimensionality* problem is the most critical reason for the construction of hierarchical fuzzy systems. Because the logical complication and the increase on the computational cost cause the system to become an inapplicable solution, and these factors unavoidably compel the expert to prefer the use of hierarchical system. In that point, expert is obligated to model subsystems according to the connections of input parameters, and generate fuzzy rules by considering this connectivity. At first glance, generation of the fuzzy rules for the subsystem which directly receives original inputs seems easier. However this task may be more difficult and even impossible for the inner layers. Because establishing an association between the outputs of subsystems become highly challenging. This problem may have a solution by generating subsets of the rule set that belongs to single fuzzy inference system, yet the generation of rules for single system is another challenge that cannot be overcome for every problem. Consequently the expert makes the decision about hierarchic structure for easy generation of hierarchical rules. As a result, are critical points about hierarchical modeling like its subjectivity (depends on the expert knowledge), and its structure which varies according to the problem inevitably. Nevertheless, neither a constant strategy for fuzzy rules and membership functions nor a standard architecture can be valid or accurate for the solution of all prediction problems. At that point, an investigation on 'how possible to ensure the accuracy of the chosen hierarchic structure' becomes essential and worth to discuss. In general hierarchical modeling, in question, comprises the organization of the subsystems and the variables in the hierarchical system are as follows: division of input space to the subsystems, membership functions, rules determined for each subsystem, the order of inputs, number of layers and subsystems, and the distribution of the subsystems into the layers due to the architectural purposes etc. However the only issue that is considered here is the architectural (or structural) modeling.

Different structures whose most critical aim is providing the most accurate and stable outputs have been already proposed in various studies [3], and clearly classified in Chung and Duan's study [4] as *aggregated*, *incremental* and *cascaded*. In *incremental* structures each layer has only one subsystem. In addition each inner-layer subsystem receives both one or more original inputs and the output(s) of its predecessor which is located in lower layer. Regarding the *aggregated* structures the layers are allowed to contain more than one subsystem. However, original inputs should be linked to the subsystems in the lowest layer. In the *cascaded* structure, on the other hand, all of the inputs are linked to the lowest layer and the output of this layer is linked to the successive one in a layer-by-layer manner. Since all of the original inputs are received by one subsystem, the *curse of dimensionality* problem is not able to be overcome by this

structure. It is possible to commentate that any other hierarchic structure is a customized version of one of these classes. Wei and Wang [5] proposed an alternative *incremental* structure in which each subsystem has two inputs in order to perform the prediction with minimum number of rules. This binary separation is also discussed in [6] as Alternative Model of Hierarchical Fuzzy System (*AHFS*) to minimize the computational cost. In this study this minimized rule based is designed by a tree-like structure by placing two subsystems in each layer.

Hierarchical fuzzy systems are implemented into several research since they were proposed by Raju *et al.* [2]. Ramirez and Mayorga [7], in example, used hierarchical fuzzy systems based on *incremental* structure to generate online portals which are customized for the particular user. This system has two connected subsystems to classify the customers, and for the identification of the customers' interests. Cheong's study [8] is an example in which an *incremental* structure is used for the solution of forecasting foreign exchange rates. Another *incremental* structure is proposed by Neogi *et al.* [9] for performance appraisal of university non-teaching staff. In this study five sub-problems (communication, motivation, interpersonal, decision making abilities and knowledge level) modeled by individual blocks, and in each block a few fuzzy systems are operated parallel. Daftaribesheli *et al.* [10], on the other hand, utilized from a *cascaded* structure to develop Fuzzy Slope Mass Rating system which calculates the Basic Rock Mass Rating and adjusting factors in separated subsystems, and combines the outputs. Mahapatra *et al.* [11] used this type of structure for water quality prediction in Indian. In fact these examples can be easily extended [12][13][14][15][16]. Ultimately the common critical drawback of these studies which is essentially addressed here is their insistence on using one constant hierarchic structure and not making any evaluation by using different ones. Therefore it is highly infeasible to ensure the accuracy of the chosen structure unless the alternative structures are eliminated by accuracy based comparisons. Different hierarchic structures have been proposed as mentioned previously. Nevertheless there is not an efficient solution to adapt a standard structure into every problem. For example Wei and Wang's [5] minimized rule based or Rattasiri and Halgamuge's *AHFS* [6] can be reasonably recommended because it minimizes the cost of computation. However it is possible for this structure to become inapplicable and/or inaccurate for the current problem because these structures increases the number of layers and makes the inner layer rule generation to be obstructed. On the other hand, more accurate outputs are likely to be provided with any other structure. Therefore it is certain that the hierarchic structure varies depend on the problem itself, and the question of which structure provides the most accurate solution is ambiguous. Furthermore the chosen operations, especially the defuzzification strategy, during the inference are another point to be considered. Because it is a well-accepted situation that each defuzzification in the inner layers degenerates the fuzziness during the hierarchical inference process [14][17]. Therefore if it is unavoidable to defuzzify the inner layer results by any reason, the selection of defuzzification strategy should also be another research area.

It is possible to attain a few studies on fuzzy modeling specialized on hierarchical fuzzy systems. Fuzzy modeling problem is explained clearly in [18] as a method for representing nonlinear input-output relations using fuzzy rules. In other words they do not address the issue on structural purposes but the modeling of fuzzy rules or optimized design of membership functions etc. In [18] the division of variable space into subsystems of a hierarchic structure is discussed and a tree-type division is proposed for the solution. For a similar problem, Cheong [8] proposed an automated and optimized system by Evolutionary Algorithm, and Wang *et al.* [19] proposed a scheme generates rules by using Gradient-Decent Method. Besides the modeling issues, some studies examined the hierarchic order of input variables. Wang *et al.* [20] research the idea that the input variables should be ordered by their relative importance. In addition to this consideration, Rattasiri and Halgamuge [6][14] proposed to place closely bonded input pairs in the same subsystem. Although these semi-automated systems are specialized for the hierarchical fuzzy modeling, each one is examined and tested on a constant structure, and none of them considered the effects on the outputs while these systems are implemented on different structures.

Herein an experimental research is employed where different hierarchic structures for the varying number of input parameters are implemented into two logical problems: AND, XOR. The evaluated hierarchic structure types are *aggregated*, *incremental* and *AHFS* [6] respectively. In addition three most reasonable and well-known defuzzifiers are implemented into these structures to observe the effect of defuzzification in the system's output on constant structures. The accuracy is measured by Root Mean Square Error (RMSE) and R Square (R^2) by basing the single system's outputs. Because it is expected from a hierarchical fuzzy system that it should both handle the bottleneck of *curse of dimensionality*, and provide same or very close outputs with single system [17].

2 Method

This section briefly explains the utilized hierarchic structures and defuzzifiers. These factors may be thought as unrelated with each other, but in fact there is a common reason that causes the variation on the output after one of these factors is revised: the degeneration of fuzzy information during the hierarchical inference flow. This bottleneck is addressed in [14][3] and explained in [17]. Briefly degeneration, in question is mainly caused from the defuzzification step since it reduces the fuzzy data to a crisp value, and in the next layer the provided crisp value is mapped to the fuzzy data for a second time. However this fuzzified data is very likely to differ from the original one because the preservation of fuzziness cannot be accomplished due to the defuzzification step. Consequently, it is affirmed that more cognizable results can be achieved by using different defuzzifiers in the meantime of the examination that focuses on structural variations.

2.1 Hierarchical Structures

This paper addresses three hierarchic structures: *aggregated* [4], *incremental* [4], and *AHFS* [6]. The mechanism of *incremental* structure is very likely to human perception about decision making [4]. A problem is firstly considered as a small group of important patterns and an approximate solution is reached by this group. Then one or more other parameters can be combined this consideration until the final decision is made. Accordingly each subsystem except the subsystem in the lowest layer has at least one inner layer value delivered from its predecessor as the approximate solution, and one original input which also affects the final decision making. The reasoning in these subsystems is based on the provided inner layer value as well as the original input [4].

The *aggregated* structure, on the other hand, is considered as classification combination, mixture of experts modeling, and bagging in [4]. In *aggregated* methodology, in decision making, multiple decisions are made individually, and these results are combined to provide the final decision. As a hierarchical fuzzy system the original inputs are fed to the lowest layer subsystems independently, and the provided value from the inner layer subsystem is transferred to the next layer. In the following layers it is approached to the final decision by combining the inner layer outputs with each other.

AHFS [6] is also examined as a customized combination of *aggregated* structure. In this structure, each subsystem is fed by two inputs, and the hierarchical system is designed by the *incrementall*y linked subsystems. In such cases where the number of inputs is an odd number, this structure becomes a combination of *aggregated* and *incremental* structures due to the linkage of the last remaining input to the subsystem in the topmost layer.

Figure 1 illustrates an eight inputs case of *aggregated*, *AHFS* and *incremental* structures respectively when each subsystem (SFS) is modeled by 2 inputs and 1 output. Herein the number of fuzzy rules r to be determined to create a complete rule base is presented in 1 where s is the number of subsystems in hierarchy, n_i is the number of inputs fed to the i^{th} subsystem, and f_{ij} is the number of fuzzy sets belonging to j^{th} input of i^{th} subsystem. Please note that the hierarchical systems in Figure 1 have equal number of subsystems, and the only varied factor which reasons the structures to differ is the links between subsystems. In fact these structures are customized versions of minimized rule base.

$$r = \sum_{i=1}^{s} \prod_{j=1}^{n_i} f_{ij}. \tag{1}$$

2.2 Defuzzification Methods

In a Mamdani style fuzzy inference [21] four basic steps are performed to procure the output: fuzzification, rule-fitting, aggregation, and defuzzification. In fuzzification step, the original inputs are fuzzified with the membership functions belonging to related linguistic variable, and the following two operations are performed on fuzzy data such as association of the fuzzified input with the

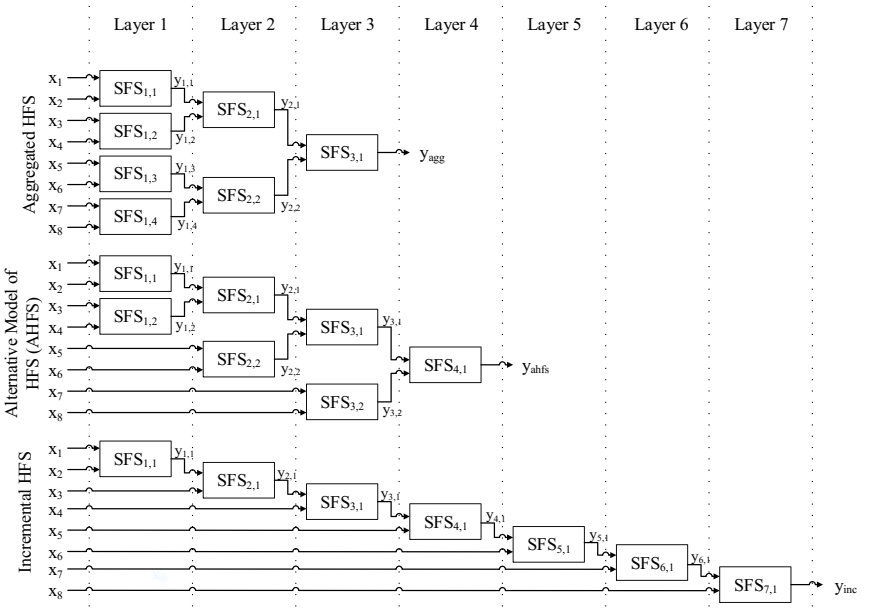


Fig. 1. Sample hierarchical fuzzy systems from each type of hierarchic structures

predetermined rules, and producing fuzzy decision. Then the defuzzification is performed to convert fuzzy result to crisp value. However fully expression of this fuzzy data with a single crisp value and the formula used here is highly challenging and cannot be standardized for every inference problem. Thus several defuzzifiers have been proposed, and making some trial inferences with different types of defuzzification methods are supposed to determine which one is most appropriate for the problem.

Herein three of well-known defuzzifiers are used such as Center of Area (COA), Bisector of Area (BOA) and Weighted Average (WA). The brief explanation of them is as follows: (*i*) COA defuzzifier returns the center of the region which obtained from the triggered membership functions of output variable. (*ii*) BOA calculates the line which divides this region into two equal pieces. (*iii*) WA method returns an average value based on the centroid of each membership function of output variable which is weighted with its maximum triggering degree.

3 Experiments

Experiments are performed on the problems of logical AND and XOR gates. At first single fuzzy inference systems are built to solve both of the experimental problems. Then three hierarchic structures by using three different defuzzifiers

are also built, and total nine different hierarchical systems are obtained. There are several reasons for the selection of logical gates as experimental subjects: (*i*) they are simple, transparent and well known cases which do not need to be specialized to understand its implementation (*ii*) rule generation is objective since the effect of expert knowledge is eliminated (*iii*) modeling the subsystems is possible and practically standardized, but only costly in such cases where the number of input parameters is increased (*iv*) it is possible to solve the problem by single fuzzy system besides the hierarchical ones. Last factor is much more essential because the primary evaluation criterion of the hierarchical fuzzy systems' accuracy is the output provided from equivalent single fuzzy system. The main expectation from a hierarchical system is to obtain close or similar results with the single system outputs otherwise significancy of building a hierarchical system becomes open to discussion. Because these systems is not proposed to provide more accurate results than corresponding single system, but to eliminate the logical and cost-wise bottlenecks while modeling.

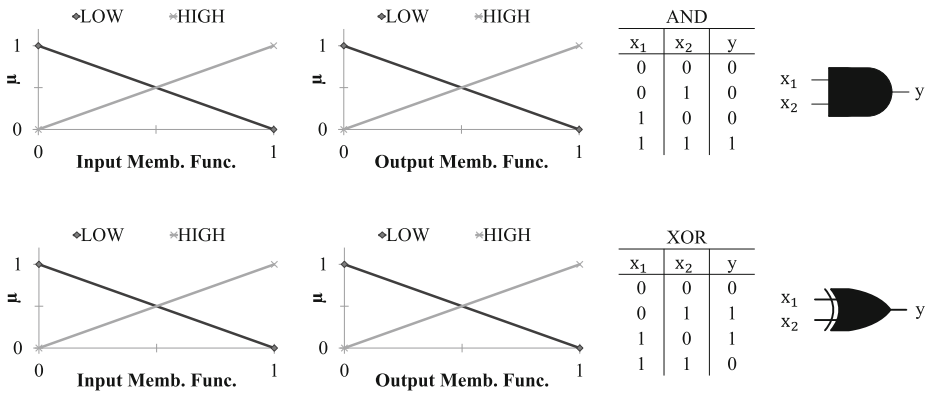


Fig. 2. The membership functions and rules for logical cases

The single fuzzy system and hierarchical fuzzy systems are implemented by using three structures (*aggregated*, *incremental* and *AHFS*) for the varying number of input parameters (four, six and eight). The structures for the case of eight inputs have been shown in Figure 1. Please note that the structural difference between *aggregated* and *AHFS* is only seen when the number of inputs is eight. Herein the number of subsystems is seven, and it is being constant, even if the structure is changed. Each subsystem is fed by two inputs since this separation minimizes the number of rules (four rules for each subsystem). As shown in Figure 2, each input has two fuzzy sets expressed by linear membership functions with the range of [0..1] and, linguistically named as *low* and *high*, respectively.

Three data sets are automatically produced to evaluate the constructed fuzzy systems. These sets consist of input tuples contains four, six and eight inputs and each variable in the input tuples varies between five values in the range from

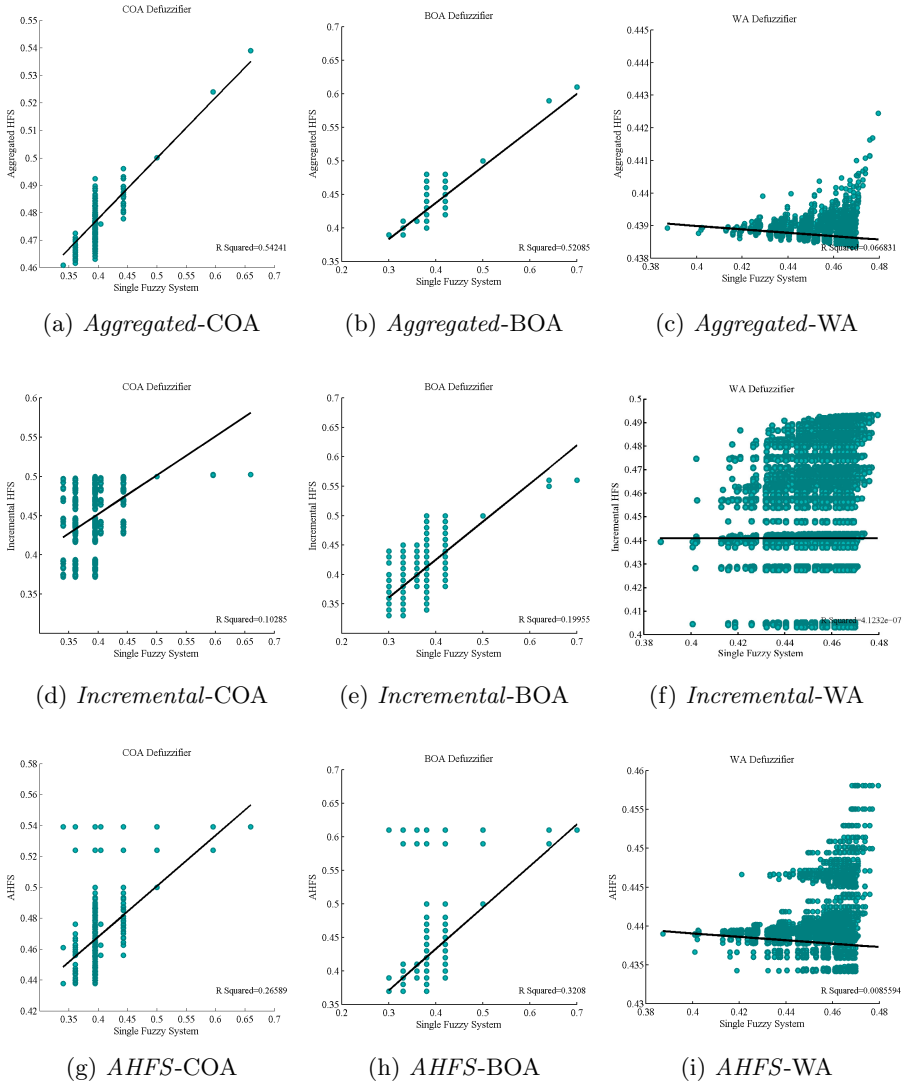
0.1 to 0.9 by the increase of 0.2. Therefore, the total number of the data set used in testing is 5^4, 5^6 and 5^9.

4 Results and Discussion

The most critical assumption addressed in this study is the differentiation on the hierarchic structures causes the outputs to be varied, and the existing applications do not realize this differentiation since they utilize from a constant structure. Another factor that creates again differentiation on the outputs is the defuzzification steps employed in the inner layers of hierarchical system because defuzzification steps performed in each subsystem changes the fuzziness level during the hierarchical inference. In fact different defuzzifiers create this change in different quantity, since each one generalizes the fuzzy data into a single value by using discrete and different strategies. Therefore, the effect of using different defuzzifiers is also investigated in the experiments.

Before the presentation of the experimental results, it may be better to point out the Figure 3 in detail. Presented results are obtained for AND case with eight inputs and each scatterplot shows the differentiation of one hierarchical system's output to the equivalently modeled single system's output. In addition a trend line is presented to observe the distribution. This plots can be evaluated to see the effect of changing defuzzification methods and the hierarchic structures separately. Herein the WA defuzzifier seems to be inconvenient to this problem. It may likely be because of the asymmetry of output membership functions since this defuzzifier provides more convenient outputs in the use of symmetric membership functions. Regarding the hierarchic structure, on the other hand, the *aggregated* structure seems closer to the related single system. It is clear that none of the hierarchical fuzzy systems can provide the single systems' outputs regardless of the chosen structure or defuzzifier. More importantly, differently modeled hierarchical systems in Figure 3 do not show similar or even close behaviors to each other's. In addition, there is a serious differentiation on the distribution of provided outputs, and this differentiation cannot be handled by using different defuzzifiers.

RMSE and R^2 values are presented in Table 1 and Table 2 for AND and XOR cases respectively. Herein the first and second columns identify the utilized structures (*aggregated*,*incremental* and *AHFS*) and the number of input parameters which symbolized as n. The following column tuples, on the other hand, shows the RMSE and R^2 values calculated by basing the output of single fuzzy system, and the subtraction of RMSE values from R^2. By this subtraction, two performance criteria are fused to make comparisons easier. Table 1 and Table 2 can be evaluated horizontally and vertically to see the effect of changed hierarchic structure and defuzzifier respectively. From the horizontal aspect, for both AND and XOR cases, the *aggregated* structure provides the closest outputs to its equivalent single system regardless the utilized defuzzifier. From the vertical aspect, while in AND case results, the BOA defuzzifier approaches to single system accuracy more than its counterparts, the same observation cannot



(a) *Aggregated*-COA (b) *Aggregated*-BOA (c) *Aggregated*-WA

(d) *Incremental*-COA (e) *Incremental*-BOA (f) *Incremental*-WA

(g) *AHFS*-COA (h) *AHFS*-BOA (i) *AHFS*-WA

Fig. 3. Comparisons for differently constructed hierarchical systems' outputs with single system outputs for AND gate with eight inputs

Table 1. Accuracy based performance comparison for AND case

Struc.	n	COA RMSE	COA R^2	COA R^2-RMSE	BOA RMSE	BOA R^2	BOA R^2-RMSE	WA RMSE	WA R^2	WA R^2-RMSE
Agg.	4	0.056	0.923	0.867	0.035	0.928	0.893	0.012	0.733	0.721
Inc.	4	0.055	0.450	0.395	0.039	0.779	0.740	0.024	0.189	0.165
AHFS	4	0.056	0.923	0.867	0.035	0.928	0.893	0.012	0.733	0.721
Agg.	6	0.071	0.499	0.428	0.046	0.666	0.620	0.014	1E-4	-0.014
Inc.	6	0.064	0.186	0.122	0.048	0.397	0.349	0.027	0.004	-0.023
AHFS	6	0.071	0.499	0.428	0.046	0.666	0.620	0.014	1E-4	-0.014
Agg.	8	0.082	0.542	0.460	0.051	0.521	0.470	0.019	0.067	0.048
Inc.	8	0.067	0.103	0.036	0.052	0.110	0.058	0.030	4.01E-7	-0.030
AHFS	8	0.073	0.266	0.193	0.050	0.321	0.271	0.020	0.0086	-0.011

Table 2. Accuracy based performance comparison for XOR case

Struc.	n	COA RMSE	COA R^2	COA R^2-RMSE	BOA RMSE	BOA R^2	BOA R^2-RMSE	WA RMSE	WA R^2	WA R^2-RMSE
Agg.	4	0.033	0.999	0.966	0.021	0.999	0.978	0.010	0.993	0.983
Inc.	4	0.047	0.956	0.909	0.032	0.990	0.958	0.015	0.973	0.958
AHFS	4	0.033	0.999	0.967	0.021	0.999	0.978	0.010	0.993	0.983
Agg.	6	0.037	0.984	0.947	0.026	0.997	0.971	0.003	0.961	0.958
Inc.	6	0.046	0.950	0.904	0.035	0.995	0.960	0.005	0.927	0.922
AHFS	6	0.037	0.984	0.947	0.026	0.997	0.971	0.003	0.961	0.958
Agg.	8	0.030	0.999	0.969	0.021	0.999	0.978	1E-4	0.985	0.985
Inc.	8	0.039	0.949	0.910	0.036	0.993	0.957	0.001	0.866	0.865
AHFS	8	0.041	6.7E-34	-0.041	0.074	9E-35	-0.074	0.002	1.4E-32	-0.002

be made for XOR results, since WA defuzzifier is very close to eliminate the superiority of BOA.

By utilizing the findings of the experiments, a highly confusing question occurs: how plausible to model a hierarchical fuzzy system with a constant structure and/or defuzzifier (as the existing studies did), unless a constant model which brings optimized or at least the most reasonable results is able to be set for even a simple AND/XOR case? The authors' assessment is that: when the problem becomes more complicated, construction of a single system becomes impossible, or not worth the effort. Therefore the expert choses the hierarchic structure so as the rule generation to be easy, and does not consider the variation of the structures as different rules has to be generated in each structural variation. Since the single system is also undetermined, difference in the provided outputs from both the single systems and the differently structured hierarchical systems cannot be evaluated. Thus either a common evaluation criterion should be developed to compare the outputs of varying hierarchic structures without requiring the construction of the single system, or the core reason of the differentiation on the outputs should be extracted and overcome.

5 Conclusion

Hierarchical fuzzy systems bring a well-accepted solution for the *curse of dimensionality* bottleneck of single fuzzy systems. Naturally there is not a standardized way to construct these systems since it should be varied according to the problem itself. Therefore different hierarchic structures can be proposed and specialized for the problem. However it is observed that the existing studies which utilized the hierarchical fuzzy systems chose one constant hierarchic structures and do not taking the other potential structures into account for no reason. Nevertheless it is an inarguable fact that the obtained result may likely to change if even one tiny structural difference is made. Therefore it is highly significant to evaluate varying structures and utilizing from the one which brings the most accurate solution.

In this study the effects of using different hierarchic structures and defuzzifiers are examined by experiments performed on AND and XOR cases with varying number of input parameters. The evaluated structures are *aggregated*, *incremental* and *AHFS* respectively. The subsystems in each structure are modeled equivalently and the same number of subsystems is combined to the construction of each structure.

Obtained results prove that each variation in the hierarchic structure causes differentiation on the systems' outputs. Certainly one of these structures can provide the closest outputs to the output of single system. Herein the *aggregated* structure brings the most accurate solution. However any other structure may perform better for any other problem. Since it is based on the problem itself, the most accurate structure has to be extracted by the experiments performed over and over. A more robust solution for this problem may be developing a method which provides equality and stability regardless the hierarchic structure. In that point the subjectivity issue may be also handled besides the experts continue constructing the hierarchical fuzzy system in a way that is easy for them.

References

1. Chen, Y., Abraham, A.: Tree-Structure based Hybrid Computational Intelligence. Intelligent Systems Reference Library, vol. 2. Springer, Heidelberg (2010)
2. Raju, G., Zhou, J., Kisner, R.: Hierarchical fuzzy control. International Journal of Control **54**, 1201–1216 (1991)
3. Wang, D., Zeng, X., Keane, J.: A survey of hierarchical fuzzy systems. International Journal of Computational Conginiton **4**, 18–29 (2006)
4. Chung, F., Duan, J.: On multistage fuzzy neural network modeling. IEEE Transactions on Fuzzy Systems **8**, 125–142 (2000)
5. Wei, C., Wang, L.X.: A note on universal approximation by hierarchical fuzzy systems. Information Science **123**, 241–248 (2000)
6. Rattasiri, W., Halgamuge, S.: Computational complexity of hierarchical fuzzy systems. In: 19th International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2000, Atlanta, vol. 2, pp. 383–387 (2000)

7. Ramirez, E., Mayorga, R.: A cascaded fuzzy inference system for dynamic online portals customization. International Journal of Electrical and Computer Engineering **2**, 300–313 (2007)
8. Cheong, F.: A hierarchical fuzzy system with high input dimensions for forecasting foreign exchange rates. In: 2007 IEEE Congress on Evolutionary Computation, pp. 1642–1647 (2007)
9. Neogi, A., Mondal, A., Mandal, S.: A Cascaded Fuzzy Inference System for University Non-Teaching Staff Performance Appraisal. Journal of Information Processing Systems **7**, 595–612 (2011)
10. Daftaribesheli, A., Ataei, M., Sereshki, F.: Assessment of rock slope stability using the Fuzzy Slope Mass Rating (FSMR) system. Applied Soft Computing **11**, 4465–4473 (2011)
11. Mahapatra, S., Nanda, S.K., Panigrahy, B.: A Cascaded Fuzzy Inference System for Indian river water quality prediction. Advances in Engineering Software **42**, 787–796 (2011)
12. Haciomeroglu, M., Laycock, R., Day, A.: Fuzzy logic controlled pedestrian groups in urban environments. In: Kallmann, M., Bekris, K. (eds.) MIG 2012. LNCS, vol. 7660, pp. 326–337. Springer, Heidelberg (2012)
13. Akyüz, H., Bingül, Z., Kizir, S.: Cascade fuzzy logic control of a single-link flexible-joint manipulator. Turkish Journal of Electrical Engineering & Computer Sciences **20**, 713–726 (2012)
14. Rattasiri, W., Halgamuge, S.: Computationally advantageous and stable hierarchical fuzzy systems for active suspension. IEEE Transactions on Industrial Electronics **50**, 48–61 (2003)
15. Rattasiri, W., Wickramarachchi, N., Halgamuge, S.K.: Sensitivity-based hierarchical controller architectures for active suspension. IEEE Transactions on Control Systems Technology **16**, 103–112 (2008)
16. Takács, M.: Multilevel fuzzy approach to the risk and disaster management. Acta Polytechnica Hungarica **7**, 91–102 (2010)
17. Mutlu, B., Sezer, E.A., Nefeslioglu, H.A.: Transition of vagueness between layers of hierarchical fuzzy systems. In: International Conference on Soft Computing, Brno, pp. 189–194 (2014)
18. Tachibana, K., Furuhashi, T.: A structure identification method of submodels for hierarchical fuzzy modeling using the multiple objective genetic algorithm. International Journal of Intelligent Systems **17**, 495–513 (2002)
19. Keane, J.: Learning for hierarchical fuzzy systems based on the gradient-descent method. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 92–99 (2006)
20. Wang, L.X.: Analysis and design of hierarchical fuzzy systems. IEEE Transactions on Fuzzy Systems **7**, 617–624 (1999)
21. Mamdani, E., Assilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. International Journal of Man-Machine Studies **7**, 1–13 (1975)

SIRMs Fuzzy Inference Model with Linear Transformation of Input Variables and Universal Approximation

Hirofumi Miyajima(✉), Noritaka Shigei, and Hiromi Miyajima

Graduate School of Science and Engineering, Kagoshima University, Korimoto, 1-21-40 Kagoshima, Japan
k3768085@kadai.jp, {shigei,miya}@eee.kagoshima-u-ac.jp

Abstract. The automatic construction of fuzzy system with a large number of input variables involves many difficulties such as large time complexity and getting stuck in a shallow and local minimum. As models to overcome them, the SIRMs (Single Input Rule Modules) and DIRMs (Double Input Rule Modules) models have been proposed. However, they are not always effective in accuracy. In the previous paper, we have proposed the model composed of two phases; the first is a linear transformation of input to intermediate variables and the second is to use SIRMs model. It was shown that the proposed model is superior in accuracy and the number of rules to the conventional models in numerical simulation. In this paper, we will show theoretically that the proposed model is a universal approximator. Further, in order to show the effectiveness of the proposed model, numerical simulation will be performed.

Keywords: Fuzzy inference system · SIRMs model · DIRMs model · Universal approximation

1 Introduction

Many studies on fuzzy inference systems and their learning methods have been made [1–3]. The aim of these studies is to construct self-tuning fuzzy systems from given data based on the steepest descend method in many cases. Obvious drawbacks of the method are the lack of interpretability and getting stuck in a shallow local minimum. Further, there is a high dimensional problem that the number of fuzzy rules increases with increasing of input variables [4,5]. The SIRMs model aims to obtain a better solution by using fuzzy inference system[6,7]. However, it is known that the SIRMs model does not always achieve good performance in non-linear problems. Therefore, the DIRMs model has been also proposed as a generalized SIRMs model, in which each module is composed of two input variables [8,9]. The ability of DIRMs model is not so high compared with the conventional model, though it is more effective than the SIRMs model and the number of parameters does not need so much compared with the conventional models. In the previous paper, we have proposed the model as

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 561–575, 2015.
DOI: 10.1007/978-3-319-19258-1_46

another generalized one composed of two phases; the first is a linear transformation of input to intermediate variables and the second is to use SIRMs model [10]. In numerical simulations, it was shown that the proposed model is superior in accuracy and the number of rules to the conventional models. Further, it can be applied to classification problems with a large number of input variables.

In this paper, we will show theoretically that the proposed model is a universal approximator by the technique used in the proof of universal approximation for three layered neural networks. Further, in order to show the effectiveness of the proposed model, numerical simulation will be performed.

2 Fuzzy Inference Model and Its Learning

2.1 Fuzzy Inference Model and Its Learning

The conventional fuzzy reasoning model using the delta rule is described [2,3]. Let $Z_j = \{1, \cdots, j\}$ for the positive integer j. Let $\boldsymbol{x} = (x_1, \cdots, x_m)$ and y be input and output data, respectively, where x_j for $j \in Z_m$ and y are real numbers. Then the rule of simplified fuzzy inference model is expressed as

$$R_i \ : \text{ if } x_1 \text{ is } M_{i1} \text{ and } \cdots \ x_m \text{ is } M_{im} \text{ then } y \text{ is } w_i, \tag{1}$$

where $i \in Z_n$ is a rule number, $j \in \ Z_m$ is a variable number, M_{ij} is the membership function of the antecedent part, and w_i is the weight of the consequent part.

A membership value of the antecedent part μ_i for input $\boldsymbol{x}$ is expressed as follows:

$$\mu_i = \prod_{j=1}^{m} M_{ij}(x_j), \tag{2}$$

where M_{ij} is the membership function of the antecedent part. Let c_{ij} and b_{ij} denote the center and the width values of M_{ij}, respectively.

The following Gaussian and triangular membership functions are used as M_{ij}:

$$M_{ij}\left(x_j\right) = \exp\left(-\frac{1}{2}\left(\frac{x_j - c_{ij}}{b_{ij}}\right)^2\right) \tag{3}$$

for Gaussian membership function.

$$M_{ki}(x) = \begin{cases} 0 \left(x < c_{ki} - \frac{b_{ki}}{2}\right) \\ \frac{2}{b_{ki}}\left[x - \left(c_{ki} - \frac{b_{ki}}{2}\right)\right] \left(c_{ki} - \frac{b_{ki}}{2} \leq x \leq c_{ki}\right) \\ -\frac{2}{b_{ki}}\left[x - \left(c_{ki} + \frac{b_{ki}}{2}\right)\right] \left(c_{ki} < x \leq c_{ki} + \frac{b_{ki}}{2}\right) \\ 0 \left(x > a_{ki} + \frac{b_{ki}}{2}\right) \end{cases} \tag{4}$$

for triangular membership function. The output y^* of fuzzy inference is calculated by the following equation:

$$y^* = \frac{\sum_{i=1}^{n} \mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \tag{5}$$

In order to construct the effective model, the conventional learning is introduced. The objective function E is defined to evaluate the inference error between the desirable output y^r and the inference output y^*.

In this section, we describe the conventional learning algorithm [2,4,5]. Let $\boldsymbol{D} = \{(x_1^p, \cdots, x_m^p, y_p^r) | p \in Z_P\}$ be the set of learning data. The objective of learning is minimizing the following mean square error(MSE):

$$E = \frac{1}{P} \sum_{p=1}^{P} (y_p^* - y_p^r)^2. \tag{6}$$

In order to minimize the objective function, many learning methods for parameters proposed. In this paper, learning method based on the steepest descent method is introduced. In order to minimize the objective function E, the parameter $\alpha \in \{c_{ij}, b_{ij}, w_i\}$ is updated based on the steepest descent method as follows:

$$\alpha(t+1) = \alpha(t) - K_\alpha \frac{\partial E}{\partial \alpha} \tag{7}$$

where t is iteration time and K_α is a constant. When Gaussian membership function eq.(3) is used, the following relation holds.

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{i=1}^n \mu_i} \cdot (y^* - y^r) \cdot (w_i - y^*) \cdot \frac{x_j - c_{ij}}{b_{ij}^2} \tag{8}$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \cdot (y^* - y^r) \cdot (w_i - y^*) \cdot \frac{(x_j - c_{ij})^2}{b_{ij}^3} \tag{9}$$

$$\frac{\partial E}{\partial w_i} = \frac{\mu_i}{\sum_{i=1}^n \mu_i} \cdot (y^* - y^r) \tag{10}$$

The conventional learning algorithm is shown below[2].

Learning Algorithm A

1 [Initialization]: The initial number of rules, c_{ij}, b_{ij} and w_j are set randomly. The threshold θ_1 for inference error is given. Let T_{max} be the maximum number of learning times. The learning coefficients K_c, K_b and K_w are set.
2 : Let $t = 1$.
3 : Let $p = 1$.
4 : An input and output data $(x_1^p, \cdots, x_m^p, y_p^r)$ is given.
5 : Membership value of each rule is calculated by Eqs.(2) and (3).
6 [Output of fuzzy inference]: Inference output y_p^* is calculated by Eq.(5).
7 [Updating parameter]: Real number w_j is updated by Eq. (10).
8 [Termination]: Parameters c_{ij} and b_{ij} are updated by Eqs.(8) and (9).
9 : If $p = P$ then go to the next step. If $p < P$ then $p \leftarrow p + 1$ and go to Step 4.
10 : Inference error $E(t)$ is calculated by Eq.(6). If $E(t) \leq \theta_1$ then learning is terminated.
11 : If $t \neq T_{max}$ then $t \leftarrow t+1$ and go to Step 3. Otherwise learning is terminated.

2.2 SNIRMs and SIRMs Models

SNIRMs and SIRMs models are introduced [8,9]. Let U_k^m be the set of all ordered k-tuples of Z_m, that is

$$U_k^m = \{l_1 \cdots l_k | l_i < l_j \text{ if } i < j\}. \tag{11}$$

Then, each rule of SNIRMs model for U_k^m is defined as follows:

$$\begin{aligned}\text{SNIRM}-l_1 \cdots l_k : \{R_i^{l_1 \cdots l_k} : \text{if } x_{l_1} \text{ is } M_i^{l_1} \text{ and } \cdots \text{ and } x_{l_k} \text{ is } M_i^{l_k} \\ \text{then } y_{l_1 \cdots l_k} \text{ is } w_i^{l_1 \cdots l_k}\}_{i=1}^n\end{aligned} \tag{12}$$

Example 1. For $U_1^4 = \{1, 2, 3, 4\}$, the obtained system is as follows:

$$\begin{aligned}&\text{SNIRM} - 1 : \{R_i^1 : \text{if } x_1 \text{ is } M_i^1 \text{ then } y_1 \text{ is } w_i^1\}_{i=1}^n \\ &\text{SNIRM} - 2 : \{R_i^2 : \text{if } x_2 \text{ is } M_i^2 \text{ then } y_2 \text{ is } w_i^2\}_{i=1}^n \\ &\text{SNIRM} - 3 : \{R_i^3 : \text{if } x_3 \text{ is } M_i^3 \text{ then } y_3 \text{ is } w_i^3\}_{i=1}^n \\ &\text{SNIRM} - 4 : \{R_i^4 : \text{if } x_4 \text{ is } M_i^4 \text{ then } y_4 \text{ is } w_i^4\}_{i=1}^n\end{aligned}$$

Let $\boldsymbol{x} = (x_1, \cdots, x_m)$. The fitness of the i-th rule and the output of SNIRM$-l_1 \cdots l_k$ are as follows:

$$\mu_i^{l_1 \cdots l_k} = M_i^{l_1}(x_{l_1}) M_i^{l_2}(x_{l_2}) \cdots M_i^{l_k}(x_{l_k}), \tag{13}$$

$$y_{l_1 \cdots l_k}^0 = \frac{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k} w_i^{l_1 \cdots l_k}}{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k}}. \tag{14}$$

In this model, in addition to the conventional parameters c, b and w, the importance degree h is introduced. Let h_L be the importance degree of each module $L = l_1 \cdots l_k \in U_k^m$.

$$y^* = \sum_{L \in U_k^m} h_L \cdot y_L^0 \tag{15}$$

A learning algorithm for SNIRMs (including SIRMs and DIRMs) model is given as follows[8,9]. When Gaussian membership function(3) is used, $\frac{\partial E}{\partial \alpha}$'s are calculated as follows:

$$\frac{\partial E}{\partial h_L} = (y^* - y^r) y_L^0 \tag{16}$$

$$\frac{\partial E}{\partial w_i^L} = h_L \cdot \frac{\mu_i^L}{\sum_{i=1}^n \mu_i^L} (y^* - y^r) \tag{17}$$

$$\frac{\partial E}{\partial c_i^L} = h_L \cdot (y^* - y^r) \frac{(w_i^L - y_L^0)\mu_i^L}{\sum_{i=1}^n \mu_i^L} \frac{\partial \mu_i^L}{\partial c_i^L} \tag{18}$$

$$\frac{\partial E}{\partial b_i^L} = h_L \cdot (y^* - y^r) \frac{(w_i^L - y_L^0)\mu_i^L}{\sum_{i=1}^n \mu_i^L} \frac{\partial \mu_i^L}{\partial b_i^L} \tag{19}$$

Then, learning algorithms for SNIRMs models are introduced as follows[8]:

Learning Algorithm B(k)

1 [Initialization]: The initial parameters, c_i^L, b_i^L, w_i^L, θ_1, T_{max}, K_c, K_b and K_w are set.

2 : Let $t = 1$.

3 : Let $p = 1$.

4 : An input and output data $(x_1^p, \cdots, x_m^p, y_p^r)$ is given.

5 : Membership value of each rule is calculated by Eq.(13).

6 [Output of fuzzy inference]: Inference output y_p is calculated by Eq.(15).

7 [Updating parameters]: Importance degree h_L is updated by Eq.(16).

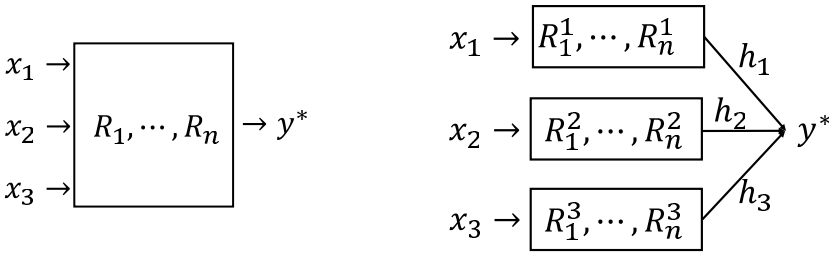
8 : Real number w_i^L is updated by Eq.(17).

9 : Parameters c_i^L and b_i^L are updated by Eqs.(18) and (19).

10 [Termination]: If $p = P$ then go to the next step. If $p < P$ then $p \leftarrow p + 1$ and go to Step 4.

11 : Inference error $E(t)$ is calculated by Eq.(6). If $E(t) < \theta_1$ then learning is terminated.

12 : If $t \neq T_{max}$, $t \leftarrow t + 1$ and go to Step 3. Otherwise learning is terminated.



(a)conventional model (b)SIRMs model

Fig. 1. The conventional and SIRMs models for $m = 3$

The cases of $k = 1$ and $k = 2$ are called SIRMs and DIRMs models, respectively[10]. Fig.1 shows the relation between the simplified fuzzy inference and SIRMs models for $m = 3$. Examples 1 is a module for SIRMs model for $m = 4$. It is known that the SIRMs model does not always achieve good performance in non-linear systems [8,9]. On the other hand, when the number of input variables is large, learning algorithm for the conventional model requires a large time complexity and tends to easily get stuck into a shallow local minimum.

2.3 Proposed Fuzzy Inference Model

The conventional fuzzy inference model shows the sufficient capability in the non-linear problems, but it needs a large number of rules (parameters). There-

fore, SIRMs model is proposed, but we cannot always get the model satisfying with sufficiently accuracy. Then, how can we construct fuzzy inference model? We have proposed a model in the previous paper [10]. The proposed model is composed of two phases: the first phase is a linear transformation of input variables and the second phase is to construct SIRMs model. The proposed method is shown in Fig.2. Let $\boldsymbol{x} = (x_1, \cdots, x_m)$ and y^* be input and output data, respectively.

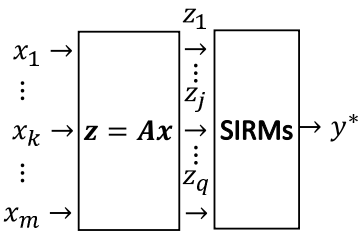


Fig. 2. The concept of proposed model

The first phase is a linear transformation $\boldsymbol{A}$ of input to intermediate variables $\boldsymbol{z} = (z_1, \cdots, z_q)^T$ as follows:

$$\begin{pmatrix} z_1 \\ \vdots \\ z_j \\ \vdots \\ z_q \end{pmatrix} = \begin{pmatrix} a_{10} \cdots a_{1k} \cdots a_{1m} \\ \vdots \\ a_{j0} \cdots a_{jk} \cdots a_{jm} \\ \vdots \\ a_{q0} \cdots a_{qj} \cdots a_{qm} \end{pmatrix} \begin{pmatrix} x_0 \\ \vdots \\ x_k \\ \vdots \\ x_m \end{pmatrix}, \tag{20}$$

where $\boldsymbol{A} = (a_{jk})$ for $j = 1, \cdots, q$, $k = 0, 1, \cdots, m$ and $z_j = \sum_{k=0}^{m} a_{jk} x_k$ $(x_0 = 1)$. The number q is selected arbitrary.

The second phase is SIRMs model with $z_1, \cdots, z_q$. The output y^* is calculated as follows:

$$\mu_i^j = M_i^j(z_j), \tag{21}$$

$$y_j^0 = \frac{\sum_{i=1}^{n} \mu_i^j w_i^j}{\sum_{i=1}^{n} \mu_i^j}, \tag{22}$$

$$y^* = \sum_{j=1}^{q} h_j y_j^0. \tag{23}$$

When Gaussian membership function is used, $\frac{\partial E}{\partial \alpha}$'s are calculated as follows:

$$\frac{\partial E}{\partial h_j} = (y^* - y^r) y_j^0, \tag{24}$$

$$\frac{\partial E}{\partial w_i^j} = h_j \frac{\mu_i^j}{\sum_{i=1}^n \mu_i^j} (y^* - y^r), \tag{25}$$

$$\frac{\partial E}{\partial c_i^j} = (y^* - y^r) h_j \frac{(w_i^j - y_j^0)\mu_i^j}{\sum_{i=1}^n \mu_i^j} \frac{z_j - c_i^j}{(b_i^j)^2}, \tag{26}$$

$$\frac{\partial E}{\partial b_i^j} = (y^* - y^r) h_j \frac{(w_i^j - y_j^0)\mu_i^j}{\sum_{i=1}^n \mu_i^j} \frac{(z_j - c_i^j)^2}{(b_i^j)^3}. \tag{27}$$

Further, $\frac{\partial E}{\partial a_{jk}}$ is as follows:

$$\frac{\partial E}{\partial a_{jk}} = (y^r - y^*) \frac{h_j}{\sum_{i=1}^n \mu_i^j} \sum_{i=1}^n \mu_i^j (w_i^j - y_j^0) \frac{z_j - c_i^j}{\left(b_i^j\right)^2} \tag{28}$$

A learning algorithm for the proposed model is given as follows [10]:

Learning Algorithm C

Let θ and T_{max} be the threshold for the inference error and the maximum number of learning, respectively.

1 [Initialization]: The initial parameters, c_{ij}, b_{ij}, w_{ij}, h_j, a_{jk}, θ, T_{max}, K_c, K_b, K_w and K_a are set. Let $t = 1$.

2: Let $p = 1$.

3 : An input and output data $(x_1^p, \cdots, x_m^p, y_p^r) \in \boldsymbol{D}$ is given.

4: From the Eq.(20), input vector $(z_1^p, \cdots, z_l^p)$ is obtained.

5 [Output of fuzzy inference]: Output of each rule is calculated from Eqs.(21) and (22). Inference output y_p^* is calculated from Eq.(23).

6 [Updating parameters]: Importance degree h_j is updated by Eq.(24). Real number w_{ij} is updated by using the Eq.(25). Parameters c_{ij} and b_{ij} are updated by Eqs.(26) and (27) respectively. a_{jk} is updated by Eq.(28).

7 [Termination]: If $p = P$ then go to the next step. If $p < P$ then $p \leftarrow p + 1$ and go to Step 3.

8: Inference error $E(t)$ is calculated by Eq.(6). If $E(t) < \theta$ then learning is terminated.

9: If $t \neq T_{max}$, $t \leftarrow t + 1$ and go to Step 2. Otherwise learning is terminated.

Remark that the number l is determined according to the threshold θ_1 for inference error.

Example 2. Let us explain the relation between SIRMs and proposed models using EX-OR problem with 2 variables [2,10]. The EX-OR problem is defined as follows:

$$y = x_1 \oplus x_2, \tag{29}$$

where $x_1, x_2, y \in \{0, 1\}$ and $\oplus$ means the Exclusive OR operation.

When Learning Algorithm is performed to the problem, the following fuzzy inference model with $q = 1$ and $n = 3$ is obtained:

$$
\begin{aligned}
& z = 0.39 - 0.49x_1 + 0.71x_2, && (30)\\
& h_1 = -1.10,\\
& if\ z\ is\ M_1^1\ then\ y\ is\ -0.98,\\
& if\ z\ is\ M_2^1\ then\ y\ is\ 0.33,\\
& if\ z\ is\ M_3^1\ then\ y\ is\ -0.97,\\
& M_1^1(z) = \exp\left(-\frac{(z+0.01)^2}{0.49}\right), && (31)\\
& M_2^1(z) = \exp\left(-\frac{(z-0.50)^2}{0.49}\right), && (32)\\
& M_3^1(z) = \exp\left(-\frac{(z-1.00)^2}{0.49}\right). && (33)
\end{aligned}
$$

Eq.(30) shows the linear transformation of x_1 and x_2. Eqs.(31), (32), and (33) show three membership functions (attributes) for z. In this case, the centers of membership functions are regarded as three diagonal lines shown in Fig.3. As a result, the output for data $(0, 1)$ or $(1, 0)$ nearly equals to 1 and output for input $(0, 0)$ or $(1, 1)$ nearly equals to 0, because the small numbers of weights are assigned to the rules for Lines 1 and 3. On the other hand, we can draw only any line parallel to x_1-axis or x_2-axis in the case of SIRMs model. Therefore, we cannot assign the same value to $(0, 1)$ and $(0, 1)$. That is, we can not constract any SIRMs model performing Eq.(29)[8,9]. The following result holds.

[Proposition][8]
The EX-OR problem with 2 variables cannot be implemented by any SIRMs model. □

In order to evaluate the effectiveness of the proposed model, some numerical simulations have already been performed in the previous paper[10]. In the next section, we will consider the capability of the proposed method theoretically.

3 The Proposed Fuzzy System as Universal Approximator

In order to prove the denseness of soft computing model in the set of continuous functions, the Stone-Weierstrass theorem is generally used[2,11]. The proposed model does not satisfy the Stone-Weierstrass theorem because multiplicative condition fails. Therefore, we cannot prove the denseness of the proposed model by using Stone-Weierstrass theorem. In this case, the capability of the model may be shown in constructive way by using the function analysis method. It seems that the proof method using the function analysis is the first attempt for fuzzy inference model. The approximation capabilities of the proposed model will be shown by using the same technique as Hornik et al.[2,12]. In the following, the

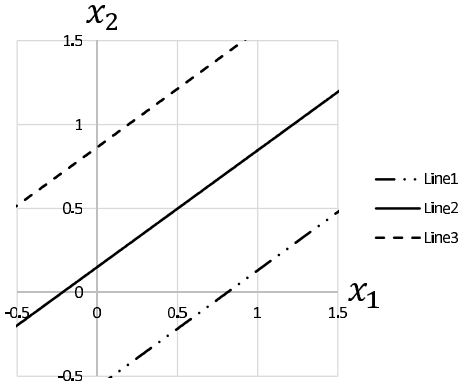


Fig. 3. The figure to explain the proposed model

denseness of the proposed model is proved. See Ref.[2] about the mathematical terms. The first step shows that any cosig function with one variable can be approximated by a proposed model, where a cosine squashing function is called a cosig function. In the second step, we prove that any cosig function can be approximated by a proposed model.

[Definition]

Let $\boldsymbol{S}$ be a compact set in $\boldsymbol{R}^m$. The set Ω is dense in $C[\boldsymbol{S}]$ iff for any $\varepsilon \in \boldsymbol{R}$ and $g \in C[\boldsymbol{S}]$, there exists a function $f \in \Omega$ such that $|g(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon$ for any $\boldsymbol{x} \in \boldsymbol{S}$. □

[Lemma][2,12]

Let $\boldsymbol{\Omega}$ be the set of all the functions that can be represented by the cosig functions on a compact set $\boldsymbol{S} \subset \boldsymbol{R}^m$ as follows:

$$\boldsymbol{\Omega}_Q = \{g(\boldsymbol{x}) = \sum_{l=1}^{Q} u_l cosig\left(\sum_{k=0}^{m} d_{lk} x_k + \theta_l\right) \\ |u_l, d_{lk}, \theta_l \in \boldsymbol{R}, \boldsymbol{x} \in \boldsymbol{S}\} \tag{34}$$

$$\boldsymbol{\Omega} = \bigcup_{Q=1}^{\infty} \boldsymbol{\Omega}_Q, \tag{35}$$

where $x_0 = 1$ and

$$cosig(x) = \begin{cases} 1 & (x > 0) \\ cos(2\pi x) & (-\frac{1}{2} \leq x \leq 0) \\ -1 & (x < -\frac{1}{2}) \end{cases} \tag{36}$$

Then $\boldsymbol{\Omega}$ is dense in $C[\boldsymbol{S}]$. □

[Theorem]

Let $\boldsymbol{\Phi}$ be the set of all the functions that can be represented by the following functions on a compact set $\boldsymbol{S} \subset \boldsymbol{R}^m$.

$$\boldsymbol{\Phi}_H = \{f(\boldsymbol{x}) = \sum_{j=1}^{H} h_j \frac{\sum_{i=1}^{n} M_i^j \left(\sum_{k=0}^{m} a_{jk} x_k\right) w_i^j}{\sum_{i=1}^{n} M_i^j \left(\sum_{k=0}^{m} a_k^j x_k\right)} \\ |h_j, a_{jk}, b_i^j, c_i^j, w_i^j \in \boldsymbol{R}, \boldsymbol{x} \in \boldsymbol{S}\} \tag{37}$$

$$\boldsymbol{\Phi} = \bigcup_{H=1}^{\infty} \boldsymbol{\Phi}_H, \tag{38}$$

where M_i^j is triangular function shown as Eq.(4)

Then $\boldsymbol{\Phi}$ is dense in $C[\boldsymbol{S}]$. □

[Proof]

From Lemma, it is sufficient to show that for any cosig function $g \in \Omega$ and any $\varepsilon > 0$ there is a function $f \in \Phi$ satisfying the following property.

$$|f(\boldsymbol{x}) - g(\boldsymbol{x})| < \varepsilon \quad \text{for any } \boldsymbol{x} \in \boldsymbol{S} \tag{39}$$

First, let us prove the case of $m = 1$. Fig.4 shows a cosig function $g(\boldsymbol{x})$ for the case of $m = 1$. Let us consider the interval $[-L, L]$ including $[-\frac{1}{2}, 0]$. Then,

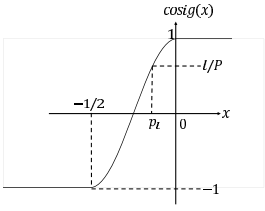


Fig. 4. The figure of a cosig function

for any $\varepsilon > 0$, we select a positive integer P satisfying $\frac{2}{P} < \varepsilon$ and the following condition;

$$p_l = \sup_{x \in [-\frac{1}{2}, 0]} \{x | cosig(x) = \frac{l}{P}\}, \tag{40}$$

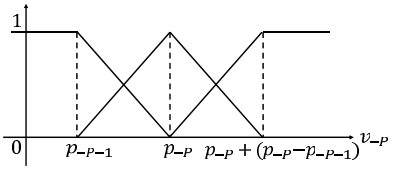
where $l = -P, \cdots, P$.

In order to approximate a function $g(x)$, let us consider a fuzzy system using triangular membership functions with $\sum_{i=1}^{n} M_i^k(v_k) = 1$ for $k = \{-P, \cdots, P\}$. That is,

$$f(x) = \sum_{k=-P}^{P} h_k \sum_{i=1}^{n} M_i^k(v_k) w_i^k \tag{41}$$

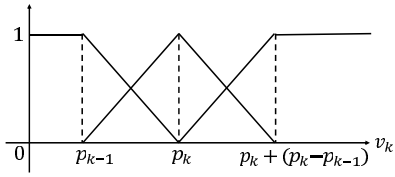
Then, SIRMs model composed of $2P + 1$ modules are constructed as Fig.5.

Let $Q = 2P + 1$ as the number of modules and $n = 3$ as the number of labels. In order to construct symmetric triangular membership functions, the centers and the widths are determined as follows:



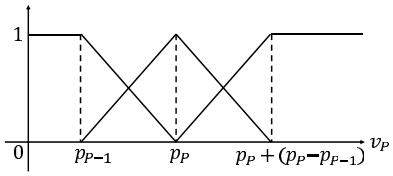
Module $-P$

$\vdots$



Module k

$\vdots$



Module P

Fig. 5. The figure to explain the $2P + 1$ modules

$$c_1^k = p_{k-1},\ c_2^k = p_k,\ c_3^k = p_k + (p_k - p_{k-1})$$
$$b_i^k = 2(p_k - p_{k-1})$$
$$h_k = 1,$$

where $k \in \{-P, \cdots, P\}$ and $p_{-P-1} = p_{-P+1} - p_{-P}$ (See Fig.5).

Further, the weights for k-th rule are determined as follows:

$$w_i^k = \begin{cases} \frac{1}{P} & \text{for } i = 2, 3 \\ 0 & \text{for } i = 1 \end{cases} \quad k \neq -P \tag{42}$$

$$w_i^{-P} = -1 \quad \text{for } i = 1, 2, 3,\ k = -P \tag{43}$$

Let $v_k = x\,(k \in \{-P, \cdots, P\})$. From the property of $\boldsymbol{\Phi}$ with $m = 1$, we can obtain the following function.

$$f(x) = \sum_{k=-P}^{P} \sum_{i=1}^{3} M_i^k(x) w_i^k \tag{44}$$

because, $\sum_{i=1}^{3} M_i^k(x) = 1$ for $k = -P, \cdots, P$ and $h_j = 1$.

Let us consider the relation between $f(x)$ and $g(x)$.

(i)$x = p_k (k = -P, \cdots, P)$

If $x = p_{-P}$ then

$$f(x) = -1 = cosig(p_{-P}) \tag{45}$$

If $x = p_k (k \neq -P)$ then

$$f(x) = -1 + \frac{P + k}{P} = \frac{k}{P} = cosig(p_l) \tag{46}$$

From Eqs.(45) and (46),

$$|f(x) - g(x)| = 0 < \varepsilon \tag{47}$$

(ii)$p_{k-1} < x < p_k (k = -P + 1, \cdots, P)$

$\frac{k-1}{P} < f(x) < \frac{k}{P}$ and $\frac{k-1}{P} < cosig(x) < \frac{k}{P}$. Therefore, it holds for

$$|f(x) - g(x)| = |\left(\frac{k-1}{P} + \frac{e_1}{P}\right) - \left(\frac{k-1}{P} + \frac{e_2}{P}\right)|$$
$$< \frac{e_1 + e_2}{P} < \frac{2}{P} < \varepsilon \tag{48}$$

(iii)$x < p_{-P},\ x > p_P$

$$x < p_{-P} : f(x) = cosig(x) = -1 \tag{49}$$
$$x > p_P : f(x) = cosig(x) = 1 \tag{50}$$

From Eqs.(49) and (50),

$$|f(x) - g(x)| = 0 < \varepsilon \tag{51}$$

Then, in the case of $m = 1$, the following holds:

$$|f(x) - g(x)| < \varepsilon \quad \text{for any } x \in S \tag{52}$$

In the second step, let us consider any function $G(\boldsymbol{x}) \in \Omega$

$$\begin{aligned} G(\boldsymbol{x}) &= \sum_{j=1}^{J} u_j cosig\left(\sum_{k=1}^{m} a_{jk} x_k + \theta_j\right) \\ &= \sum_{j=1}^{J} u_j cosig(z_j), \end{aligned} \tag{53}$$

where $z_j = \sum_{k=0}^{m} a_{jk} x_k (a_{j0} = \theta_j$ and $a_{jk} = d_{jk}$ for $k \neq 0)$.

As $z_j \in \boldsymbol{R}$, we can select within the distance $\frac{\varepsilon}{J|u_j|}$ of $cosig(z_j)$ the following function;

$$f(z_j) = \sum_{k=-P}^{P} \sum_{i=1}^{n} M_i^k(z_j) w_i^k \tag{54}$$

such that

$$|f(z_j) - g(z_j)| < \min_j \{\frac{\varepsilon}{J|u_j|}\} \tag{55}$$

Let the function $F(\boldsymbol{x})$ of linear sum composed of J pieces of $f(z_j)$

$$F(\boldsymbol{x}) = \sum_{j=1}^{J} u_j \sum_{k=-P}^{P} \sum_{i=1}^{n} M_i^k(z_j) w_i^k \tag{56}$$

By using the transformation from Eq.(41) to an element of $\boldsymbol{\Phi}$, the following relation holds:

$$F(\boldsymbol{x}) = \sum_{j=1}^{J} u_j \frac{\sum_{k=-P}^{P} \sum_{i=1}^{n} M_i^k(z_j)(2P+1) w_i^k}{\sum_{k=-P}^{P} \sum_{i=1}^{n} M_i^k(z_j)}, \tag{57}$$

where $\sum_{k=-P}^{P} \sum_{i=1}^{n} M_i^k(z_j) = 2P + 1$ for $j = 1, \cdots, J$.

Then, the following result holds:

$$\sup_{\boldsymbol{x} \in S} |F(\boldsymbol{x}) - G(\boldsymbol{x})| < \sum_{j=1}^{J} |u_j| \frac{\varepsilon}{J|u_j|} = \varepsilon \tag{58}$$

From Lemma and the result, theorem is obtained. □

The result is an existence theorem and there is another problem whether we can get effective system. Learning Algorithm C is a learning algorithm based on the steepest descent method.

4 Numerical Simulation

In order to compare the capability of Learning Algorithm C with one of Learning Algorithm A and B(k), let us show a numerical simulation for classification problem, where algorithms A, B(1), B(2) and C mean Learning Algorithms A, B($k = 1$), B($k = 2$) and C, respectively.

Iris, Wine, Sonar and BCW data from UCI database known as one of benchmark problems are used for numerical simulation[13]. They are shown in Table1. In this simulation, 5-fold cross-validation is used. The result of simulation is the average value from ten trials.

Table 1. The dataset for classification problems[13]

	Iris	Wine	Sonar	BCW
The number of data	150	178	208	683
The number of input	4	13	60	9
The number of class	3	3	2	2

In this simulation, the parameters K_c and K_b are set to 0.001, the parameters K_w, K_h and K_a are set to 0.01 and T_{max} is set to 50000, respectively. The initial conditions for c_{ij}, b_{ij}, w_{ij}, h_j and a_{ik} are *equal intervals*, $\frac{1}{4}\times$(*the domain of input*), *random on* $[0, 1]$, *random on* $[0, 1]$ and *random on* $[-1, 1]$, respectively.

Table 2. The result of simulation for Classification problem

		Iris	Wine	Sonar	BCW
A	Learning	0.000			
	Test	0.071			
B(1)	Learning	0.020	0.012	0.006	0.056
	Test	0.054	0.081	0.261	0.061
B(2)	Learning	0.001	0.002		0.000
	Test	0.059	0.062		0.070
C	Learning	0.029	0.001	0.001	0.017
	Test	0.029	0.054	0.220	0.036

(a)Missclassification rate

		Iris	Wine	Sonar	BCW
A	Time	394.8			
	♯para	(729)			
B(1)	Time	17.7	62.7	267.1	165.9
	♯para	(40)	(130)	(600)	(90)
B(2)	Time	145.0	1535.4		1506.0
	♯para	(276)	(3588)		(1656)
C	Time	13.5	20.2	37.7	71.9
	♯para	(45)	(72)	(213)	(60)

(b)The number of parameters and learning time

Table 2 shows the result of numerical simulation. In Table 2 (a) and (b), Learning, Test, Time and ♯para mean the missclassification rates for learning and test, learning time(s) and the number of parameters, respectively. The missclassification rate means the proportion of data that are incorrectly classified. In Table 2, oblique lines mean that it is difficult to simulate the problem because the number of parameters is needed too much. As a result, Table 2 shows that the proposed method is superior in accuracy to SIRMs and DIRMs and in the number of parameters and learning time to conventional fuzzy inference model.

5 Conclusions

In the previous paper, we have proposed a fuzzy inference model. It is composed of two phases: The first is a linear transformation of input variables and the second is to construct SIRMs model. From some numerical simulations, it was shown that the proposed model was superior in accuracy and the number of parameters to other models. In this paper, we have proved theoretically that the proposed model is a universal approximator by using the function analysis method. Further, in order to show the effectiveness of the proposed model, numerical simulation was performed.

In the future work, we will show the convergence of learning algorithm for the proposed model and propose a new learning algorithm for it.

References

1. Lin, C., Lee, C.: Neural Fuzzy Systems. Prentice Hall, PTR (1996)
2. Gupta, M.M., Jin, L., Homma, N.: Static and Dynamic Neural Networks. IEEE Press (2003)
3. Casillas, J., Cordon, O., Herrera, F., Magdalena, L.: Accuracy Improvements in Linguistic Fuzzy Modeling. STUDFUZZ, vol. 129. Springer, Heidelberg (2003)
4. Nomura, H., Hayashi, I., Wakami, N.: A learning method of simplified fuzzy reasoning by genetic algorithm. In: Proc. of the Int. Fuzzy Systems and Intelligent Control Conference, pp. 236–245 (1992)
5. Fukumoto, S., Miyajima, H.: Learning Algorithms with Regularization Criteria for Fuzzy Reasoning Model. Journal of Innovative Computing Information and Control **2**(1), 249–263 (2006)
6. Yubazaki, N., Yi, J., Hirota, K.: SIRMS (Single Input Rule Modules) Connected Fuzzy Inference Model. J. Advanced Computational Intelligence **1**(1), 23–30 (1997)
7. Yi, J., Yubazaki, N., Hirota, K.: A Proposal of SIRMs Dynamically Connected Fuzzy Inference Model for Plural Input Fuzzy Control. Fuzzy Sets and Systems **125**, 79–92 (2002)
8. Miyajima, H., Shigei, N., Miyajima, H.: Fuzzy Inference Systems Composed of Double-Input Rule Modules for Obstacle Avoidance Problems. IAENG International Journal of Computer Science **41**(4), 222–230 (2014)
9. Miike, S., Miyajima, H., Shigei, N., Noo, K.: Fuzzy Reasoning Model With Deletion of Rules Consisting of Small-Number-of-Input Rule Modules. Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, 621–629 (2010)
10. Miyajima, H., Shigei, N., Miyajima, H.: A proposal of SIRMs model with linear transformation of input variables. In: SCIS and ISIS 2014, pp. 311–316 (2014)
11. Wang, L.X., Mendel, J.M.: Fuzzy Basis Functions, Universal Approximation and Orthogonal Least Square Learning. IEEE Trans. Neural Networks **3**(3), 807–814 (1992)
12. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks and Universal Approximators. Neural Networks **2**, 359–366 (1989)
13. UCI Repository of Machine Learning Databases and Domain Theories. ftp://ftp.ics.uci.edu/pub/machinelearning-Databases

A New Approach of Fuzzy Neural Networks in Monthly Forecast of Water Flow

Ruben Araújo (✉), Mêuser Valença, and Sérgio Fernandes

Polytechnic School of Pernambuco, University of Pernambuco, Recife, Brazil
{rna,meuser}@ecomp.poli.br,
smmfast@gmail.com

Abstract. The water influences many areas of society. Energy production, own consumption, and irrigation make use of this resource. Within the electricity production context, the flow forecasting process of the rivers that feed the electricity generation plants is very important for the success of this type of generation. Historically, neural networks have been highlighted in this type of application, in particular, the Multilayer Perceptron. Fuzzy neural networks have also been used for the same purpose. Our goal in this paper is to propose the hybridization of a fuzzy neural network that makes use of Multilayer Perceptron architecture with the Least Squares Method, to the improvement the process of monthly forecast of water flow. The neuro fuzzy network is compared to a Multilayer Perceptron network Classic through experiments and statistical tests. The results showed improvements in predictive process in most cases, suggesting that the new approach has significant potential application.

Keywords: Forecast of water flow · Fuzzy neural network · Multilayer Perceptron

1 Introduction

Water is one of the most important natural resources for humans. Applications in energy production, own consumption, irrigation, and flood forecasting in cities influence the quality of social life. According to the WWF (World Wildlife Fund) [1], by the year 2025 about two-thirds of the world population may face problems related to water scarcity if it remains the current rate of consumption. According to Davies [2], in 2013 the floods were responsible for 44% of deaths from natural disasters.

In Brazil, water is fundamental in the electricity production process. ANEEL (National Electric Energy Agency) [3] information show that, currently, approximately 63% of the energy produced in the country is generated from hydroelectric plants. This water potential benefits from rivers with high water flow and the rugged terrain in favor of this type of energy generation.

In this scenario, it is very important that water flow estimates of the rivers that feed these plants are provided in the short, medium, and long term. Prior identification of possible situations of low energy production because of low water flow in rivers enables alternative actions can be planned to supply the energy demand.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 576–586, 2015.
DOI: 10.1007/978-3-319-19258-1_47

In this context, improvements in energy planning are directly related to the quality of the forecasting process.

Historically, the neural network has been successful in forecasting water flow applications. In particular, the MLP (Multilayer Perceptron) [4,5] has been widely used for this purpose.

Other types of neural networks have also been used in forecast of water flow are the fuzzy neural networks. According to Kasabov [6], one of the first researchers to contribute in this area, fuzzy neural networks are recommended in forecasting and classification applications as they enable the union of improvements in the accuracy of the result (fuzzy logic) and the generalizability of the same (the neural network). Since then, the literature suggests the use of fuzzy neural networks in the water flow forecasting context [7,8], including with self-adaptive models called ANFIS (Adaptive Neuro Fuzzy Inference System) [9,10] and dynamic models, also known as DENFIS (Dynamic Evolving Neuro-Fuzzy Inference System) [11].

Therefore, our goal in this work is to improve the process of monthly estimates of water flow from the hybridization of a fuzzy neural network (which uses architecture of an MLP) with the LSM (Least Squares Method) [12]. We had the idea of using the LSM to update the network weights between the hidden and output layers. Finally, a Classical MLP network (simple MLP) is tested and compared with the Fuzzy MLP (new proposal) in order to assess the improvements obtained with this new approach.

This paper is divided into several sections, from the Introduction. In Section 2, is seen a review of key concepts of fuzzy neural networks and the proposed new approach. In the Sections 3 and 4, we describe respectively, the configuration of the experiments and the results that were obtained. Finally, in Section 5 we show the conclusions reached.

2 Background

The purpose of this section is to describe the important concepts related to fuzzy neural networks and the contributions that were incorporated the technique.

2.1 Fuzzy Neural Networks

Neural networks that use fuzzy logic are known as fuzzy neural networks. Basically, two new information are aggregated in this type of technique: a step of fuzzification / defuzzification and rule engine [6].

The fuzzification step consists in assigning a grade of relevance to each input value of the neural network. Typically, the relevance is between values from 0 to 1. They are calculated by applying a membership function for the value of each input. Literature reports some examples of these functions, and the Triangular, Gaussian, and Logistic functions the best known [13]. Regarding defuzzification, there are some metrics that can be used successfully, and whose mechanism depends on the context of the problem to be solved. The Center of Gravity method and the Averaging method are examples [13].

On the testing phase, we saw that the functions of fuzzification Γ-membership, Exponential-like [13], and a variation of Logistic produced the best results and for this reason, we chose for the experiments. They are respectively described in (1), (2), and (3).

$$f(x_i) = 1 - e^{-\frac{(x_i - mean_i)^2}{(deviation_i)^2}} \tag{1}$$

$$f(x_i) = \frac{1}{(1 + \frac{x_i^2}{deviation_i})} \tag{2}$$

$$f(x_i) = \frac{1}{(1 + e^{-\frac{(x_i - mean_i)}{deviation_i^2}})} \tag{3}$$

Where x_i corresponds to each input of the neural network, $mean_i$ is the average of all possible input values x_i available in the database and $deviation_i$, the standard deviation of the values obtained.

The defuzzification process usually occurs at the entry of the existing neurons in the output layer of the neural network. In this case, after conducting some tests, we noticed that the Center of Gravity and Averaging methods provided more quality in the predictive process, so being chosen. Equation (4) shows the Center of Gravity method while the (5) summarizes the Averaging method.

$$netout_i = \frac{\sum_{j=0}^{m} w_{ij} . f_{hid}(x_j)}{\sum_{j=0}^{m} f_{hid}(x_j)} \tag{4}$$

$$netout_i = \frac{\sum_{j=0}^{m} w_{ij} . f_{hid}(x_j)}{m+1} \tag{5}$$

Where w_{ij} is the weight of the connection of neurons of the output layer with neurons of the hidden layer, $f_{hid}(x_j)$ is the value assigned to each neuron j in the hidden layer by applying the activation function itself and $m+1$ is the number of neurons in the hidden layer.

Another important concept of fuzzy neural networks is the notion of rules. According Kasabov [6], a rule matches the decision to generate an action by observing behavior (s). In mathematical terms, rules can be described as:

If (x_1 is A_1) and (x_2 is B_1) and ... and (x_n is N_1) then (y is Z_1)
If (x_1 is A_2) and (x_2 is B_2) and ... and (x_n is N_2) then (y is Z_2)
...
If (x_1 is A_t) and (x_2 is B_t) and ... and (x_n is N_t) then (y is Z_t),

where n is the number of behaviors that are evaluated in the respective rule. The vector $(x_1, x_2, ..., x_n)$ represents the behavior of these variables. The $(A_1, A_2, ..., A_t)$, $(B_1, B_2, ..., B_t)$, ..., $(N_1, N_2, ..., N_t)$ are the values that the variable can admit. Finally, $(Z_1, Z_2, ..., Z_t)$ is the vector of actions to be taken upon the occurrence of each respective rule and t is the number of rules. Usually, in a fuzzy neural network, the rules are inferred or deduced.

Fig. 1 illustrates an example of fuzzy neural network. Through it, we note that the fuzzification stage happens after the variable values are received in the input layer by applying a membership function $f(x_i)$. The defuzzification, in turn, usually occurs in the output layer.

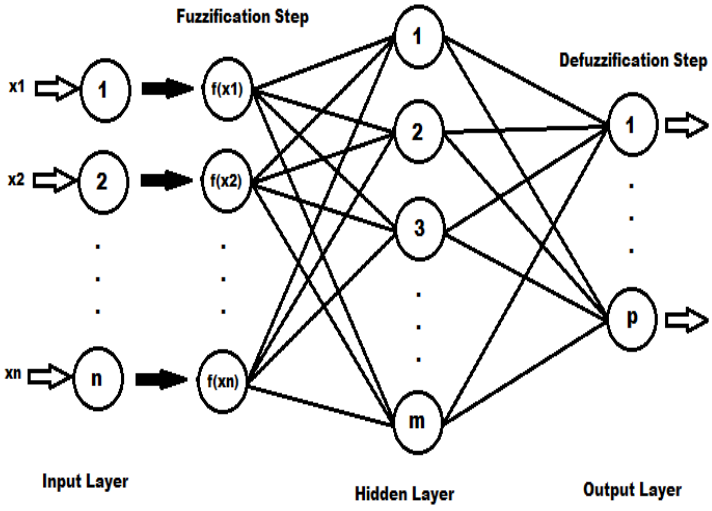


Fig. 1. Example of fuzzy neural network that makes use of MLP architecture

2.2 Hybridization of the Least Squares Method with Fuzzy Neural Networks

As previously mentioned, our goal in this work is to improve the quality of monthly forecast process of water flow through the use fuzzy neural networks. For this, the fuzzy neural network makes use of MLP architecture, in view of the good results obtained by this type of network in this context.

The Fuzzy MLP operates similarly to Classical MLP network, except for incremental steps of fuzzification / defuzzification described in subsection 2.1. The additional step is the hybridization of LSM [12] with the Fuzzy MLP network. It consists in modifying the method of updating the network weights between the hidden and output layers. So, instead of using the backpropagation, we use the LSM to update these weights. This idea is based on how the fuzzy neural network ANFIS [9,10,14] works.

The LSM is to minimize the square error between the real value and the value observed in a measurement [12]. Through this mathematical feature, we can do an analogy with how the backpropagation works in a MLP network to replace the update mechanism of the weights between the hidden and output layers. Equations (6) and (7) describe the update method of weights using the LSM.

$$[w_{new}] = (([x].[x])^{-1}.[x].[netout_{real}]) + [w_{old}] \quad (6)$$

$$[netout_{real}] = [-\ln((1/out_{real}) - 1)] \quad (7)$$

The term $[w_{new}]$ is the weight matrix to be updated, where $[w_{old}]$ is the matrix of old values. The matrix $[x]$ corresponds to the output values of the neurons in the hidden layer. In turn, the term $[netout_{real}]$ is calculated from $[out_{real}]$, which are the real flow values available in the neural network training cycle. The hybridization occurs only at this stage. Once all training data are needed to update the weights, the update using hybridization occurs in batch mode and not online.

3 Simulation Setup

The purpose of this section is the description of relevant topics for understanding the experimental arrangements.

3.1 Database

The data that were used in the experiments for the comparison of the techniques Fuzzy MLP and Classical MLP correspond to four flow series from four important Brazilian rivers for hydroelectric generation. These rivers feed the hydroelectric plants of Tucuruí, Jupiá, Tres Marias, and Foz do Areia. The data were obtained through a specialist in the field of water resources, a of the co-authors of this paper. The data are monthly measurements of water flow from January 1970 to December 2003 (408 samples in m^3/s) for each flow series.

3.2 Preprocessing Data

Typically, a step that precedes the application of intelligent techniques is the preprocessing of data that will be used. This is necessary because in some cases, missing or inconsistent data may harm the quality expected of the technique being used.

Generally, is common apply normalization in the data before the training phase in neural networks. In the context of water flow forecast, it is common also, we apply standardization in the initial data.

The standardization considers the influence of the seasonality in some types of water flow series. In this case, because the measurements are monthly, in fact, the flow tends to approach of a characteristic average in each month. Through the available historical data, we can calculate the mean and standard deviation of flow for each

of the twelve months of the year. Equation (8) shows the formula used to standardize all data. The $mean_i$ and $deviation_i$ terms are, respectively, the mean and standard deviation for the month of x_i value being standardized.

$$x_{sta} = \frac{x_i - mean_i}{deviation_i} \tag{8}$$

In addition to the standardization process, a very important step within the flow forecasting context is the normalization of data before applying the techniques. This prevents the network provided values compromise its performance due to suffer large variations. Equation (9) describes how we apply the standards for each value:

$$x_{norm} = (b-a).\frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} + a\,, \tag{9}$$

where x_i is the value to be normalized, x_{min} and x_{max}, respectively, the lowest and highest existing value related to x_i variable. The a and b terms are chosen to determine the normalization range. In this case, we chose $a=0.15$ and $b=0.85$. We made this choice to prevent the boundary region of the chosen activation function (here the interval [0;1]).

3.3 Metrics for Evaluation

One of the most common metrics to measure performance on neural networks is the MSE (Mean Square Error) [13]. For this reason, we choose it to be used in the experiments. Equation (10) shows the MSE (m^6/s^2):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{s}\sum_{j=1}^{s} e_{ij}^2\right), \tag{10}$$

where s is the number of network outputs, n is the number of instances of the database and e_{ij} is the difference between the real value and the forecast.

In order to have experiments with more consistent results, we used beyond the MSE other two metrics for evaluation: MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage Error). Equations (11) and (12) show, respectively, the MAE (m^3/s) and the MAPE (%):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{s}\sum_{j=1}^{s}\left|e_{ij}\right|\right) \tag{11}$$

$$MAPE = 100.\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{s}\sum_{j=1}^{s}\left|\frac{e_{ij}}{out_{ij}}\right|\right), \tag{12}$$

where out_{ij} is the real value which is also used by e_{ij}.

3.4 Parameters of Fuzzy MLP and Classical MLP Networks

The neural networks used in the experiments were set up using the same parameters in order to make the analysis of the results as fairly as possible. The parameterization was made from the finest results collected in exhaustive tests on Fuzzy MLP and Classical MLP networks.

The numbers of neurons of the input, hidden and output layers are, respectively, 14, 15, and 12. The architecture used for Fuzzy MLP network is the same in Fig. 1. The same figure also applies to Classical MLP network, except for existence of the steps of fuzzification and defuzzification.

The learning rate used in the two techniques was 0.25 while the momentum was 0.20. The chosen activation function for both in the hidden and output layers was the Sigmoid Logistics, which has performed well in the backpropagation algorithm.

The training methodology used in the experiments consisted of separate databases in training proportions (50%), cross-validation (25%), and test (25%). In this case, the interruption of training to find the optimal weights configuration on each neural network happened when the cross-validation MSE value in each iteration increased by three times. The initial iteration of this growth is responsible for these optimal weights.

The method of updating the weights in online mode was chosen for the Classical MLP network due to better results in this configuration. In the Fuzzy MLP network, which is the new proposal, the online mode is maintained only in the update of the weights between input and hidden layers. Between the hidden and output layers, once was used the LSM to update the weights, the way was used in batch.

In order to have statistical significance in the experiments, we chose to perform 30 simulations for each technique. The results are presented in tables and statistical tests.

4 Results

As previously mentioned, the experiments in this study are intended to compare the proposed technique Fuzzy MLP with Classical MLP. To do so, the four databases described in subsection 3.1 were used, by making use of the metrics in subsection 3.3.

The configuration of Fuzzy MLP varied in each database. For Tucuruí flow series, the fuzzification process made use of the Γ-membership function and the defuzzification, the Center of Gravity method. In the data of Foz do Areia, fuzzification and

Table 1. MSE results of the Fuzzy MLP and Classical MLP networks

Database	MSE (m^6/s^2)	
	Fuzzy MLP	***Classical MLP***
Foz do Areia	**388,051.60** (26,025.33)	442,245.26 **(19,728.92)**
Tucuruí	**94,218,946.71** (2,590,826.12)	97,992,216.88 **(1,566,738.32)**
Tres Marias	**340,760.18** (15,398.13)	351,765.19 **(9,053.02)**
Jupiá	**12,559,690.80 (189,924.09)**	12,757,725.66 (691,275.04)

defuzzification used, respectively, the variation of the Logistic function and the Averaging method. Finally, at flow series of Jupiá and Tres Marias, the fuzzification used the Exponential-like function and defuzzification, the Averaging method. All these functions and methods are described in subsection 2.1.

The Table 1, Table 2, and Table 3 summarize the results obtained. The methodology is to calculate the mean and the standard deviation (value in brackets) of the 30 simulations for each of the metrics used.

The Table 1 relate to MSE metric. Through it, we can note that the Fuzzy MLP achieved better average than the Classical MLP in all databases, however, with a standard deviation of higher values (except in Jupiá). We can note high MSE values, especially in Tucuruí, since the MSE metric is applied to the 12 outputs of each neural network. The Table 2 shows the results using the MAE. It is observed that the behavior of results is similar to that obtained by MSE in Table 1.

Table 2. MAE results of the Fuzzy MLP and Classical MLP networks

Database	**MAE (m^3/s)**	
	Fuzzy MLP	***Classical MLP***
Foz do Areia	**466.78** (14.29)	474.17 **(11.26)**
Tucuruí	**7,480.87** (92.37)	7,593.37 **(39.57)**
Tres Marias	**414.04** (15.88)	431.95 **(8.09)**
Jupiá	**2,798.94** **(31.80)**	2,820.69 (104.23)

Finally, in Table 3, we have the results obtained by MAPE metric. In this case, the Fuzzy MLP was better in the most of results. The Fuzzy MLP achieved better results in Tucuruí, Tres Marias, and Jupiá flow series while the Classical MLP was better in Foz do Areia.

Table 3. MAPE results of the Fuzzy MLP and Classical MLP networks

Database	**MAPE (%)**	
	Fuzzy MLP	***Classical MLP***
Foz do Areia	72.70 (6.48)	**58.85** **(0.79)**
Tucuruí	**109.29** (3.35)	116.36 **(1.47)**
Tres Marias	**90.50** (8.26)	102.30 **(3.41)**
Jupiá	**45.55** **(1.54)**	46.76 (2.37)

As described in subsection 3.4, we use statistical tests in the experiments in order to have more relevance in the results obtained. Due to the nature of the data found, we chose to use the Wilcoxon statistical test for comparison of Fuzzy MLP and Classical MLP techniques. This test makes no assumption of normality of samples involved. The R software was used to apply the Wilcoxon statistical test. The confidence

interval used was 95%. The Table 4 shows the results obtained with tests using the data of the 30 simulations with the metrics involved in the experiments.

The results of the statistical tests showed that Fuzzy MLP got more quality prediction in all metrics for Tucuruí and Tres Marias flow series. As for the Foz do Areia flow series, Fuzzy MLP was better using the MSE and MAE metrics, while the Classical MLP won using MAPE. Finally, in Jupiá flow series there was no statistically significant difference in using either of two techniques tried, regardless of the metric.

Table 4. Wilcoxon test results for Fuzzy MLP and Classical MLP networks

Database	MSE	MAE	MAPE
Foz do Areia	Fuzzy MLP was better than Classical MLP	Fuzzy MLP was better than Classical MLP	Classical MLP was better than Fuzzy MLP
Tucuruí	Fuzzy MLP was better than Classical MLP	Fuzzy MLP was better than Classical MLP	Fuzzy MLP was better than Classical MLP
Tres Marias	Fuzzy MLP was better than Classical MLP	Fuzzy MLP was better than Classical MLP	Fuzzy MLP was better than Classical MLP
Jupiá	There was no difference statistics significant between Fuzzy MLP and Classical MLP	There was no difference statistics significant between Fuzzy MLP and Classical MLP	There was no difference statistics significant between Fuzzy MLP and Classical MLP

Fig. 2 shows the forecast curve for Tres Marias flow series seeing 1 month ahead. Through it, we can see that the Fuzzy MLP tried to get closer to peak real flows more that the Classical MLP.

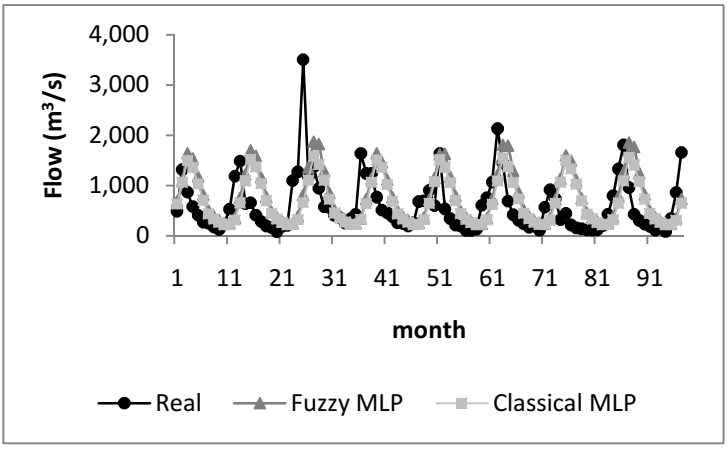


Fig. 2. Forecast curve 1 month ahead of Tres Marias flow series

5 Conclusion

This paper presented a new approach to fuzzy neural networks in monthly forecast of water flow. The proposal was the hybridization of a fuzzy neural network with LSM (Least Squares Method) to update the weights between the hidden and output layers. This proposal (Fuzzy MLP network) was compared with a Classical MLP network. At the end of this study, some informations was obtained.

The results of the experiments including application of the statistical tests showed that the Fuzzy MLP obtained higher quality in prediction process that Classical MLP network in most results. The fact of this technique have been better in more simple database (Tucuruí and Tres Marias) for prediction, in all metrics applied, suggests that the inclusion of LSM added greater precision in predictive process. It is possible that this new approach is recommended in simple flow series.

However, only these results are not yet sufficient to deeper conclusions. We noted that although the Fuzzy MLP network has obtained good performance in minimizing the error in the metrics used, the standard deviation achieved in some instances had higher values than obtained by the Classical MLP network. Certainly, further studies should be made in order to solve this problem.

As future works, we hope to compare the new proposal with other water flow forecasting techniques.

References

1. World Wildlife Fund. Washington, DC. http://www.worldwildlife.org/threats/water-scarcity (accessed January 20, 2015)
2. Davies, R.: World Disasters Report – Most Deaths Caused by Floods. http://floodlist.com/dealing-with-floods/world-disasters-report-100-million-affected-2013 (accessed January 20, 2015)
3. National Electric Energy Agency. Brasilia, Brazil. BIG – Banco de Informações de Geração – Capacidade de Geração do Brasil. http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm (accessed January 20, 2015)
4. Cruz, M.F.M., Rodrigues, L.D., Versiani, B.R.: Previsão de Vazões com a Metodologia DPFT e com Redes Neurais Artificiais. Revista Brasileira de Recursos Hídricos **15**(1), 121–132 (2010)
5. Deka, P., Chandramoulli, V.: Fuzzy Neural Network Model for Hydrologic Flow Routing. Journal of Hydrologic Engineering (July, August 2005). doi:10.1061/(ASCE)1084-0699(2005)10:4(302)
6. Kasabov, N.K.: Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. A Bradford Book, London (1996)
7. Lopes, M.S., Luna, I., Ballini, R., Soares, S.: Previsão de Vazões para o Planejamento da Operação Energética do SIN. In: IV Simpósio Brasileiro de Sistemas Elétricos, vol. 1, Goiânia, Brazil (2012)
8. Alvisi, S., Franchini, M.: Fuzzy neural networks for water level and discharge forecasting with uncertainty. Environmental Modelling & Software **26**, 523–537 (2011). doi:10.1016/j.envsoft.2010.10.016

9. Lohani, A.K., Kumar, R., Singh, R.D.: Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. Journal of Hydrology **442–443**, 23–35 (2012)
10. Firat, M., Güngör, M.: River flow estimation using adaptive neuro fuzzy inference system. Mathematics and Computers in Simulation **75**, 87–96 (2007). doi:10.1016/j.matcom.2006.09.003
11. Song, Q., Kasabov, N.: Dynamic Evolving Neuro-Fuzzy Inference System (DENFIS): Online learning and Application for Time-Series Prediction. Journal IEEE Transactions on Fuzzy Systems Conference **10**(2), 144–154 (2002)
12. Farebrother, R.W.: Linear Least Squares Computations. Marcel Dekker Inc., New York (1988)
13. Engelbrechet, A.P.: Computational Intelligence: an introduction, 2nd edn. Wiley, New Jersey (2007)
14. Chang, F.J., Chang, Y.T.: Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. Advances in Water Resources **29**(1), 1–10 (2006)

Ordering Relations Over Intuitionistic Fuzzy Quantities

Elena Mielcova(✉)

Department of Informatics and Mathematics, Silesian University in Opava, School of Business Administration in Karvina, Univerzitni namesti 1934/3, 733 40 Karvina, Czech Republic
mielcova@opf.slu.cz

Abstract. This article discuss basic concepts of relations between two intuitionistic fuzzy numbers - namely an approach based on deterministic representation, an approach based on intuitionistic fuzzy values, as well as the intuitionistic fuzzy relation. The intuitionistic fuzzy relation definition is based on the extension principle and derived with respect to expected relation properties.

Keywords: Relation operator · Intuitionistic fuzzy sets · Extension principle

1 Introduction

The incorporation of uncertainty into calculations by utilization of fuzzy sets started in 1960's, when Zadeh published his now famous article "Fuzzy Sets" [15]. Zadeh in his work expected that every fuzzy set is composed of elements characterized by a membership degree to the set. Later on, Atanassov (1986) [1] extended Zadeh's idea into the theory of intuitionistic fuzzy sets by the idea of indecisiveness – in intuitionistic fuzzy sets, both the membership degree and non-membership degree is assigned to each of their elements. Since then, the theory of intuitionistic fuzzy sets was studied and applied in various areas of mathematics and computer science. Similarly as in the fuzzy set theory, the theory of intuitionistic fuzzy sets covers also definition and operations over intuitionistic fuzzy quantities as well as over special subset of intuitionistic fuzzy quantities — intuitionistic fuzzy numbers (defined and discussed for example in [2], [3], or [5]).

One of the basic operations in the theory of intuitionistic fuzzy quantities is an ordering relation between two quantities. In order to compare two intuitionistic fuzzy quantities or numbers, there exist several definitions for ordering relations (for example [2], [12] and more). Usually these relations corresponds to specific needs in particular applications. The main aim of this text is to discuss several possible orderings of intuitionistic fuzzy quantities with respect to their basic properties.

The next text is organized as follows; the preliminaries cover basic terms from the theory of intuitionistic fuzzy sets, quantities, and numbers. The third section is devoted to orderings of fuzzy quantities. The conclusion followed by references ends the text.

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 587–596, 2015.
DOI: 10.1007/978-3-319-19258-1_48

2 Preliminaries

Atanassov (1986) [1] defined an intuitionistic fuzzy set as follows: Let a set $X = \{x_1, x_2, \ldots, x_n\}$ be fixed. Then an intuitionistic fuzzy set is defined as a set of triples $A = \{\langle x_i, \mu_A(x_i), \nu_A(x_i)\rangle; x_i \in X\}$ where functions $\mu_A : X \to L$ and $\nu_A : X \to L$ for $L = [0,1]$ define the degree of membership and the degree of nonmembership of the element $x_i \in X$ to $A \subset X$, respectively. For an intuitionistic fuzzy set, the condition $0 \leq \mu_A(x_i) + \nu_A(x_i) \leq 1$ holds for all $x_i \in X$. Value $\pi_A(x_i) = 1 - \mu_A(x_i) - \nu_A(x_i)$ is called a degree of uncertainty.

Elementary arithmetic operations over intuitionistic fuzzy sets are derived with respect to so-called extension principle with some level of generality. Extension principle over intuitionistic fuzzy sets was introduced by Çoker (1997)[4] in his article discussing a topology of intuitionistic fuzzy sets. The definition based on works of Marinov (2014) [9] and Stoeva (1999) [10]:

Suppose mapping f from X to Y usually written as $f : X \to Y$, where Domain$(f) = X$, and Range$(f) \subset Y$. The extension principle allows f to be extended to $f : IFS(X) \to IFS(Y)$, where $IFS(X)$ and $IFS(Y)$ stay for the intuitionistic fuzzy set over universes X and Y, respectively. Then for every $A \in IFS(X)$ there exist $f(A) \in IFS(Y)$ such that for every $y \in Y$

$$\mu_{f(A)}(y) = \begin{cases} \sup\{\mu_A(x); x \in f^{-1}(y)\} & \text{if } y \in \text{Range}(f) \\ 0 & \text{if } y \notin \text{Range}(f) \end{cases} \tag{1}$$

$$\nu_{f(A)}(y) = \begin{cases} \inf\{\nu_A(x); x \in f^{-1}(y)\} & \text{if } y \in \text{Range}(f) \\ 1 & \text{if } y \notin \text{Range}(f) \end{cases} \tag{2}$$

The inverse mapping f^{-1} is defined such that for any $B \in IFS(Y)$, the membership and nonmembership function is

$$\mu_{f^{-1}(B)}(x) = \mu_B(f(x)) \quad \text{and} \quad \nu_{f^{-1}(B)}(x) = \nu_B(f(x)) \tag{3}$$

for every $x \in X$.

The main aim of this work is to summarize and review basic properties of ordering relations over intuitionistic fuzzy quantities. In general, any intuitionistic fuzzy subset A of the set of real numbers R is called an intuitionistic fuzzy quantity with membership function $\mu_A : R \to [0,1]$, and nonmembership function $\nu_A : R \to [0,1]$ iff

(a) $\exists x_0 \in R$ such that $\mu_A(x_0) = 1$ and $\nu_A(x_0) = 0$
(b) $\exists x_1, x_2 \in R, x_1 < x_2$ such that for all $x \notin [x_1, x_2]$ there is $\mu_A(x) = 0$ and $\nu_A(x) = 1$.

Special subset of intuitionistic fuzzy quantities cover all intuitionistic fuzzy numbers, defined for example in [2], [3], and [5]. An intuitionistic fuzzy subset of the real line $A = \{\langle x, \mu_A(x), \nu_A(x)\ \rangle; x \in R\}$ is called an intuitionistic fuzzy number if:

(a) A is if-normal (there exist at least two points $x_0, x_1 \in X$ such that $\mu_A(x_0) = 1$ and $\nu_A(x_1) = 1$)

(b) A is if-convex (its membership function μ_A is fuzzy convex and its non-membership function ν_A is fuzzy concave);
(c) μ_A is upper semi-continuous and ν_A is lower semi-continuous;
(d) $A = \{x \in X; \nu_A(x) < 1\}$ is bounded.

3 Orderings of Fuzzy Quantities

Let a, b, and c be the intuitionistic fuzzy quantites. Then the ordering relations over such quantities should contain relations of $\preccurlyeq$, $\approx$, and alternatively $\prec$. Then ordering over set of intuitionistic fuzzy numbers should satisfy at least some properties equivalent to properties of complete ordering of real numbers, which are:

(a) the axiom of reflexivity:

$$\text{for all } a; \quad a \preccurlyeq a; \tag{4}$$

(b) the axiom of antisymmetry:

$$\text{for all } a \text{ and } b; \text{ if } a \preccurlyeq b \text{ and } b \preccurlyeq a \text{ then } a \approx b; \tag{5}$$

(c) the axiom of transitivity:

$$\text{for all } a, b, \text{ and } c; \ a \preccurlyeq b \text{ and } b \preccurlyeq c \text{ implies } a \preccurlyeq c; \tag{6}$$

(d) and the total-ordering axiom:

$$\text{for all } a \text{ and } b; \ a \preccurlyeq b \text{ or } b \preccurlyeq a. \tag{7}$$

Moreover:

(e)

$$\text{for all } a \text{ and } b; \ a \preccurlyeq b \text{ and } \neg b \preccurlyeq a \text{ implies } a \prec c; \tag{8}$$

(f)

$$\text{for all } a \text{ and } b; \ a \preccurlyeq b \ \Leftrightarrow \ a \prec b \text{ or } a \approx b. \tag{9}$$

(g)

$$\text{for all } a \text{ and } b; \ a \preccurlyeq b \text{ excludes } b \prec a. \tag{10}$$

Example: All presented fuzzy relation operation will be demonstrated on three intuitionistic (triangular) fuzzy numbers a, b, and c with membership functions as follows:

$$a: \quad \mu_a(x) = \begin{cases} \frac{x}{2} & \text{for } 0 \le x \le 2 \\ 2 - \frac{x}{2} & \text{for } 2 \le x \le 4 \\ 0 & \text{otherwise} \end{cases} \qquad \nu_a(x) = \begin{cases} 1 - \frac{x}{2} & \text{for } 0 \le x \le 2 \\ \frac{x}{2} - 1 & \text{for } 2 \le x \le 4 \\ 1 & \text{otherwise} \end{cases}$$

$$b: \quad \mu_b(x) = \begin{cases} x - 3 & \text{for } 3 \le x \le 4 \\ 5 - x & \text{for } 4 \le x \le 5 \\ 0 & \text{otherwise} \end{cases} \qquad \nu_b(x) = \begin{cases} 2 - \frac{x}{2} & \text{for } 2 \le x \le 4 \\ \frac{x}{2} - 2 & \text{for } 4 \le x \le 6 \\ 1 & \text{otherwise} \end{cases}$$

$$c: \quad \mu_c(x) = \begin{cases} \frac{x}{2} - 3 & \text{for } 6 \le x \le 8 \\ 9 - x & \text{for } 8 \le x \le 9 \\ 0 & \text{otherwise} \end{cases} \qquad \nu_c(x) = \begin{cases} 4 - \frac{x}{2} & \text{for } 6 \le x \le 8 \\ x - 8 & \text{for } 8 \le x \le 9 \\ 1 & \text{otherwise} \end{cases}$$

For simplicity, in this case numbers a, and c are ordinary fuzzy numbers. •

3.1 Ordering with Respect to Intuitionistic Fuzzy Values

According to Xu and Xia (2011) [13] the pair $(\mu_\alpha(x_i), \nu_\alpha(x_i))$ is called an intuicionistic fuzzy value, simply denoted as $\alpha_i = (\mu_{\alpha_i}, \nu_{\alpha_i})$ where $\mu_{\alpha_i} \in [0,1]$, and $\nu_{\alpha_i} \in [0,1]$. Value $s_{\alpha_i} = \mu_{\alpha_i} - \nu_{\alpha_i}$ is called score degree, and value $h_{\alpha_i} = \mu_{\alpha_i} + \nu_{\alpha_i}$ is an accuracy degree of an intuitionistic fuzzy value α_i.

For comparison of any two intuitionistic fuzzy values, $\alpha_1 = (\mu_{\alpha_1}, \nu_{\alpha_1})$, and $\alpha_2 = (\mu_{\alpha_2}, \nu_{\alpha_2})$, the next two-step method based on score and accuracy degree was discussed by Xu, Yager (2006) [14] and Xu (2007) [12]:

(1) If $s_{\alpha_1} < s_{\alpha_2}$, then $\alpha_1 < \alpha_2$;
(2) if $s_{\alpha_1} = s_{\alpha_2}$, then
  (2a) if $h_{\alpha_1} = h_{\alpha_2}$ then $\alpha_1 = \alpha_2$;
  (2b) if $h_{\alpha_1} < h_{\alpha_2}$ then $\alpha_1 < \alpha_2$.

This method for comparing two intuitionistic fuzzy values satisfies properties (a)–(g). The definition can give comparison for any two intuitionistic fuzzy quantities a, and b described as sets of fuzzy values a_i and b_i, respectively. This ordering gives sets (intervals) at which the intuitionistic fuzzy values ordering is satisfied, as is shown in the model example. In the example, $a_i = (\mu_a(x_i), \nu_a(x_i))$, and $b_i = (\mu_b(x_i), \nu_b(x_i))$ for all $x_i \in R$.

Example: Score degrees and accuracy degrees of the intuitionistic fuzzy numbers a, b, and c from the previous example are:

$$a: \quad s_a(x) = \begin{cases} x-1 \text{ for } 0 \le x \le 2 \\ 3-x \text{ for } 2 \le x \le 4 \\ -1 \quad \text{otherwise} \end{cases} \qquad h_a(x) = \begin{cases} 1 \text{ for } 0 \le x \le 2 \\ 1 \text{ for } 2 \le x \le 4 \\ 1 \quad \text{otherwise} \end{cases}$$

$$b: \quad s_b(x) = \begin{cases} \frac{x}{2} - 2 \text{ for } 2 \le x \le 3 \\ \frac{3x}{2} - 5 \text{ for } 3 \le x \le 4 \\ 7 - \frac{3x}{2} \text{ for } 4 \le x \le 5 \\ 2 - \frac{x}{2} \text{ for } 5 \le x \le 6 \\ -1 \quad \text{otherwise} \end{cases} \qquad h_b(x) = \begin{cases} 2 - \frac{x}{2} \text{ for } 2 \le x \le 3 \\ \frac{x}{2} - 1 \text{ for } 3 \le x \le 4 \\ 3 - \frac{x}{2} \text{ for } 4 \le x \le 5 \\ \frac{x}{2} - 2 \text{ for } 5 \le x \le 6 \\ 1 \quad \text{otherwise} \end{cases}$$

$$c: \quad s_c(x) = \begin{cases} x-7 \text{ for } 6 \le x \le 8 \\ 17-2x \text{ for } 8 \le x \le 9 \\ -1 \quad \text{otherwise} \end{cases} \qquad h_c(x) = \begin{cases} 1 \text{ for } 6 \le x \le 8 \\ 1 \text{ for } 8 \le x \le 9 \\ 1 \quad \text{otherwise} \end{cases}$$

In this case:

$a_i < b_i$ for all $x_i \in (\frac{16}{5}, 6)$; $b_i < a_i$ for all $x_i \in (0, \frac{16}{5}]$; $a_i = c_i$ otherwise.
$a_i < c_i$ for all $x_i \in (6, 9)$; $c_i < a_i$ for all $x_i \in (0, 4)$; $a_i = c_i$ otherwise;
$b_i < c_i$ for all $x_i \in (6, 9)$; $c_i < b_i$ for all $x_i \in (2, 6)$; $a_i = c_i$ otherwise. •

Any ordering derived from above relations between a_i and b_i would not give ordering in a sense comparison of two intuitionistic fuzzy quantities, but only some kind of "dispersion" of intuitionistic fuzzy quantities over given interval. The example of such an ordering of two intuitionistic number a and b is then:

$$a \preccurlyeq b \text{ for all } x_i \in R \text{ such that } a_i \le b_i. \tag{11}$$

This ordering is not complete — condition (d) of complete ordering is not satisfied, however other conditions are valid. This definition gives only sets (intervals) at which the given ordering is satisfied. Equivalent form of this ordering relation for fuzzy numbers is called an implicative ordering and was described in [8].

Example: Results of such an ordering for intuitionistic fuzzy numbers a, b, and c from the previous examples:

$a \prec b$ for all $x_i \in (\frac{16}{5}, 6)$; $b \prec a$ for all $x_i \in (0, \frac{16}{5}]$; $a \approx c$ otherwise.
$a \prec c$ for all $x_i \in (6, 9)$; $c \prec a$ for all $x_i \in (0, 4)$; $a \approx c$ otherwise;
$b \prec c$ for all $x_i \in (6, 9)$; $c \prec b$ for all $x_i \in (2, 6)$; $a \approx c$ otherwise. •

3.2 Ordering Using Deterministic Representation

In the least complicated example of ordering of two values is the case, when there is no intersection between membership and non-membership degrees different from basic levels $\mu(x) = 0$ and $\nu(x) = 1$. Then the ordering of two intuitionistic numbers a, and b can be written as $a \preccurlyeq b$ iff for all $x, y \in R$ such that $(\mu_a(x) \neq 0$, or $\nu_a(x) \neq 1)$, and $(\mu_b(y) \neq 0$, or $\nu_b(y) \neq 1)$ is always $x \leq y$.

This ordering is not complete, condition (d) of complete ordering is not satisfied, however all other conditions are valid, if the relations exist.

Example: This kind of ordering for intuitionistic fuzzy numbers a, b, and c from previous examples:

- $\mu_a(x)$ or $\nu_a(x)$ are different from basic levels (that means $\mu_a(x) \neq 0$ or $\nu_a(x) \neq 1$) for $x \in (0; 4)$, $\mu_c(y)$ or $\nu_c(y)$ are different from basic levels for $y \in (6; 9)$; for all possible values $x < y$, therefore $a \prec c$;
- $\mu_b(x)$ or $\nu_b(x)$ are different from basic levels for $x \in (2; 6)$, $\mu_c(y)$ or $\nu_c(y)$ are different from basic levels for $y \in (6; 9)$; for all possible values $x < y$, therefore $b \prec c$;
- $\mu_a(x)$ or $\nu_a(x)$ are different from basic levels for $x \in (0; 4)$, $\mu_b(y)$ or $\nu_b(y)$ are different from basic levels for $y \in (2; 6)$; there is no ordering relation between a, and b. •

Ordering using deterministic representation expect setting of a crisp value representing intuitionistic fuzzy number, and consequently compare these crisp values using crisp relations. Examples of such values could be modal values, or extreme possible values [8]. In the case of intuitionistic fuzzy sets it is reasonable to use score degree, which combines both membership and non-membership degree instead of solely membership degree. Then we can define:

- Ordering using modal values:

$$a \preccurlyeq b \text{ iff } \sup\{x; s_a(x) = 1\} \leq \sup\{x; s_b(x) = 1\} \tag{12}$$

- Ordering using ε-modal values:

$$a \preccurlyeq b \text{ iff for } \varepsilon \in (0, 2) \ \sup\{x; s_a(x) \geq 1 - \varepsilon\} \leq \sup\{x; s_b(x) \geq 1 - \varepsilon\} \tag{13}$$

– Ordering using extreme possible values:

$$a \preccurlyeq b \text{ iff } \sup\{x; s_a(x) > -1\} \leq \sup\{x; s_b(x) > -1\} \tag{14}$$

– Ordering using ε-modal values:

$$a \preccurlyeq b \text{ iff for } \varepsilon \in (-1;1) \ \sup\{x; s_a(x) > \varepsilon\} \leq \sup\{x; s_b(x) > \varepsilon\} \tag{15}$$

All four types of ordering satisfy conditions (a)–(g) equivalent to properties of complete ordering of real numbers.

Example: Ordering using modal values and ordering using extreme possible values for intuitionistic fuzzy numbers a, b, and c from previous examples gives the same results:

– Modal values: $\sup\{x; s_a(x) = 1\} = 2$, $\sup\{x; s_b(x) = 1\} = 4$ and $\sup\{x; s_c(x) = 1\} = 8$, therefore $a \prec b$, $a \prec c$, and $b \prec c$;
– Extreme values: $\sup\{x; s_a(x) > -1\} = 4$, $\sup\{x; s_b(x) > -1\} = 5$ and $\sup\{x; s_c(x) \geq -1\} = 9$, therefore $a \prec b$, $a \prec c$, and $b \prec c$. •

3.3 Intuitionistic Fuzzy Ordering Relation

On contrary to deterministic ordering relation, which gives exact ordering of two intuitionistic fuzzy quantities, the intuitionistic fuzzy ordering relation should give the membership and nonmembership degree of given ordering relation. This type of ordering relation can be derived from the extension principle (1):

Definition: Let $\mathbb{R}$ be the set of all intuitionistic fuzzy quantities. For $a, b \in \mathbb{R}$ we define fuzzy relations $a \preccurlyeq$ and $a \approx b$ with membership functions $\mu_{\preccurlyeq} : \mathbb{R} \times \mathbb{R} \to [0,1]$, $\mu_{\approx} : \mathbb{R} \times \mathbb{R} \to [0,1]$ and nonmembership functions $\nu_{\preccurlyeq} : \mathbb{R} \times \mathbb{R} \to [0,1]$, $\nu_{\approx} : \mathbb{R} \times \mathbb{R} \to [0,1]$ such that:

$$\mu_{\preccurlyeq}(a,b) = \sup_{x \leq y}(\min(\mu_a(x), \mu_b(y))); \tag{16}$$

$$\nu_{\preccurlyeq}(a,b) = \inf_{x \leq y}(\max(\nu_a(x), \nu_b(y))); \tag{17}$$

$$\mu_{\approx}(a,b) = \sup_{x \in R}(\min(\mu_a(x), \mu_b(x))); \tag{18}$$

$$\nu_{\approx}(a,b) = \inf_{x \in R}(\max(\nu_a(x), \nu_b(x))). \tag{19}$$

Example: Results of comparison for intuitionistic fuzzy numbers a, b, and c given in previous examples are:

$$\text{relation } a \preccurlyeq b: \quad \mu_{\preccurlyeq}(a,b) = 1 \quad \text{and} \quad \nu_{\preccurlyeq}(a,b) = 0;$$

$$\text{relation } b \preccurlyeq a: \quad \mu_{\preccurlyeq}(b,a) = \frac{1}{3} \quad \text{and} \quad \nu_{\preccurlyeq}(b,a) = \frac{1}{2};$$

$$\begin{aligned}
\text{relation } a \approx b: &\quad \mu_{\approx}(a,b) = \frac{1}{3} \quad \text{and} \quad \nu_{\approx}(a,b) = \frac{1}{2};\\
\text{relation } a \preccurlyeq c: &\quad \mu_{\preccurlyeq}(a,c) = 1 \quad \text{and} \quad \nu_{\preccurlyeq}(a,c) = 0;\\
\text{relation } c \preccurlyeq a: &\quad \mu_{\preccurlyeq}(c,a) = 0 \quad \text{and} \quad \nu_{\preccurlyeq}(c,a) = 1;\\
\text{relation } a \approx c: &\quad \mu_{\approx}(a,c) = 0 \quad \text{and} \quad \nu_{\approx}(a,c) = 1;\\
\text{relation } b \preccurlyeq c: &\quad \mu_{\preccurlyeq}(b,c) = 1 \quad \text{and} \quad \nu_{\preccurlyeq}(b,c) = 0;\\
\text{relation } c \preccurlyeq b: &\quad \mu_{\preccurlyeq}(c,b) = 0 \quad \text{and} \quad \nu_{\preccurlyeq}(c,b) = 1;\\
\text{relation } b \approx c: &\quad \mu_{\approx}(b,c) = 0 \quad \text{and} \quad \nu_{\approx}(b,c) = 1. \qquad \bullet
\end{aligned}$$

Taking into account of basic properties of such an ordering, the of axiom of reflexivity is of the form:

$$\forall a \in \mathbb{R};\ \mu_{\preccurlyeq}(a,a) = \mu_{\approx}(a,a) = 1 \text{ and } \nu_{\preccurlyeq}(a,a) = \nu_{\approx}(a,a) = 0 \tag{20}$$

which corresponds to condition (a) equivalent to the condition of total ordering of real numbers.

Similarly, axiom of antisymmetry for any $a, b \in \mathbb{R}$:

$$\begin{aligned}
\mu_{\preccurlyeq}(a,b) \wedge \mu_{\preccurlyeq}(b,a) &= \sup_{x \le y}(\min(\mu_a(x), \mu_b(y))) \wedge \sup_{x \le y}(\min(\mu_b(x), \mu_a(y))) =\\
&= \sup_{x \le y}(\min(\mu_a(x), \mu_b(y), \mu_b(x), \mu_a(y))) =\\
&= \sup_{x \in R}(\min(\mu_a(x), \mu_b(x))) = \mu_{\approx}(a,b)
\end{aligned} \tag{21}$$

$$\begin{aligned}
\nu_{\preccurlyeq}(a,b) \wedge \nu_{\preccurlyeq}(b,a) &= \inf_{x \le y}(\max(\nu_a(x), \nu_b(y))) \wedge \inf_{x \le y}(\max(\nu_b(x), \nu_a(y))) =\\
&= \inf_{x \le y}(\max(\nu_a(x), \nu_b(y), \nu_b(x), \nu_a(y))) =\\
&= \inf_{x \in R}(\max(\nu_a(x), \nu_b(x))) = \nu_{\approx}(a,b)
\end{aligned} \tag{22}$$

which corresponds to axiom (b) of complete ordering axioms. The total ordering axiom (d) is naturally fulfilled, however transitivity axiom is not generally valid. The transitivity relations in their weak form can be given as:

$$\mu_{\preccurlyeq}(a,c) \ge \min(\mu_{\preccurlyeq}(a,b), \mu_{\preccurlyeq}(b,c)) \tag{23}$$

$$\nu_{\preccurlyeq}(a,c) \le \max(\nu_{\preccurlyeq}(a,b), \nu_{\preccurlyeq}(b,c)) \tag{24}$$

Next example demonstrates how transitivity relations (23, 24) in some cases can be fulfilled and in specific cases can be broken.

Example: Results of comparison for intuitionistic fuzzy numbers a, b, and c given in previous examples are:

$$\text{relation } b \preccurlyeq a: \quad \mu_{\preccurlyeq}(b,a) = \frac{1}{3} \quad \text{and} \quad \nu_{\preccurlyeq}(b,a) = \frac{1}{2};$$

$$\text{relation } a \preccurlyeq c: \quad \mu_{\preccurlyeq}(a,c) = 1 \quad \text{and} \quad \nu_{\preccurlyeq}(a,c) = 0;$$
$$\text{relation } b \preccurlyeq c: \quad \mu_{\preccurlyeq}(b,c) = 1 \quad \text{and} \quad \nu_{\preccurlyeq}(b,c) = 0.$$

Transitivity relations for membership and nonmembership degree functions in weak form are valid:

$$1 = \mu_{\preccurlyeq}(b,c) \geq \min(\mu_{\preccurlyeq}(b,a), \mu_{\preccurlyeq}(a,c)) = \min(\frac{1}{3}, 1) = \frac{1}{3};$$

$$0 = \nu_{\preccurlyeq}(b,c) \leq \max(\nu_{\preccurlyeq}(b,a), \nu_{\preccurlyeq}(a,c)) = \max(\frac{1}{2}, 0)) = \frac{1}{2}.$$

Let's define new intuitionistic fuzzy number d such that:

$$d: \quad \mu_d(x) = \begin{cases} x+1 \text{ for } -1 \leq x \leq 0 \\ 1-x \text{ for } 0 \leq x \leq 1 \\ 0 \qquad \text{otherwise} \end{cases} \qquad \nu_d(x) = \begin{cases} -\frac{x}{2} \text{ for } -2 \leq x \leq 0 \\ \frac{x}{2} \text{ for } 0 \leq x \leq 2 \\ 1 \qquad \text{otherwise} \end{cases}$$

Results of selected comparison for intuitionistic fuzzy numbers a, b, and d are:

$$\text{relation } b \preccurlyeq a: \quad \mu_{\preccurlyeq}(b,a) = \frac{1}{3} \quad \text{and} \quad \nu_{\preccurlyeq}(b,a) = \frac{1}{2};$$

$$\text{relation } a \preccurlyeq d: \quad \mu_{\preccurlyeq}(a,c) = \frac{1}{3} \quad \text{and} \quad \nu_{\preccurlyeq}(a,d) = \frac{1}{2};$$
$$\text{relation } b \preccurlyeq d: \quad \mu_{\preccurlyeq}(b,d) = 0 \quad \text{and} \quad \nu_{\preccurlyeq}(b,d) = 1;$$

In this case, transitivity relations for membership and nonmembership degree functions in weak form are not valid:

$$0 = \mu_{\preccurlyeq}(b,d) \not\geq \min(\mu_{\preccurlyeq}(b,a), \mu_{\preccurlyeq}(a,d)) = \min(\frac{1}{3}, \frac{1}{3}) = \frac{1}{3};$$

$$1 = \nu_{\preccurlyeq}(b,d) \not\leq \max(\nu_{\preccurlyeq}(b,a), \nu_{\preccurlyeq}(a,d)) = \max(\frac{1}{2}, \frac{1}{2})) = \frac{1}{2}. \quad \bullet$$

3.4 Alternative Ordering Relations

Another possibility how to compare two intuitionistic fuzzy quantities is to consider evaluating the possibility of relation by a real number $r_{\preccurlyeq}(a,b) \in [0,1]$. Similar idea is behind the measure of similarity of two intuitionistic fuzzy sets as discussed in [6], [7] or [11].

In order to maintain the compatibility with fuzzy relations and with idea of comparison of intuitionistic fuzzy values with respect to their score degree described in [14], the possibility of relation is described as follows: For any pair of intuitionistic fuzzy numbers a, b the possibility of the relation $a \preccurlyeq b$ is defined as a real number $r_{\preccurlyeq}(a,b) \in [0,1]$ such that for $x, y \in R$:

$$r_{\preccurlyeq}(a,b) = \frac{\sup_{x \leq y}(\min(s_a(x), s_b(y))) + 1}{2} \tag{25}$$

The equivalence relation $a \approx b$ is valid with the possibility:

$$r_{\approx}(a,b) = \frac{\sup_{x,y \in R}(\min(s_a(x), s_b(y))) + 1}{2} \tag{26}$$

Logical conjunction of $a \preccurlyeq b$ and $b \preccurlyeq a$ is not $a \approx b$, but in general

$$r_{\approx}(a,b) = r_{\approx}(b,a) \geq \min(r_{\preccurlyeq}(a,b), r_{\preccurlyeq}(b,a)) \tag{27}$$

In the case of monotonic score degree, this statement is valid with equality. This type of ordering satisfy axioms of reflexivity, antisymmetry and total ordering, however axiom of transitivity, similarly in the case of intuitionistic fuzzy ordering, is not necessarily satisfied.

Example: Comparison of intuitionistic fuzzy numbers a, and b from the previous examples:

$$r_{\preccurlyeq}(a,b) = \frac{\sup_{x \leq y}(\min(s_a(x), s_b(y))) + 1}{2} = \frac{1+1}{2} = 1$$

$$r_{\preccurlyeq}(b,a) = \frac{\sup_{x \leq y}(\min(s_b(x), s_a(y))) + 1}{2} = \frac{(-\frac{1}{5}) + 1}{2} = \frac{2}{5}$$

$$r_{\approx}(a,b) = \frac{\sup_{x,y \in R}(\min(s_a(x), s_b(y))) + 1}{2} = \frac{(-\frac{1}{5}) + 1}{2} = \frac{2}{5}$$

Because of monotonicity of the score degree function for both intuitionistic fuzzy quantities a, and b, the relation $r_{\approx}(a,b) = \min(r_{\preccurlyeq}(a,b), r_{\preccurlyeq}(b,a))$ is valid with equality. •

4 Conclusions

In this paper, several possible orderings of intuitionistic fuzzy quantities were discussed. While ordering relation based on intuitionistic fuzzy values does not give reasonable ordering over all R, their importance in the theory of pattern recognition problems is irreplaceable, as shown in [6] or [7]. In the case of deterministic ordering relations based on modal and extreme values, each intuitionistic fuzzy quantity is represented by a deterministic object, and the ordering is transformed into complete ordering of real numbers. Even if theoretically this is the best approach with respect to suitability of ordering, the main disadvantage is in the loss of information given by membership and nonmembership function. This loss of information is not a case of intuitionistic fuzzy ordering relation derived from the extension principle. However, this ordering does not satisfy the axiom of transitivity. This is also the case of the last presented measure of possibility of relation evaluation.

Acknowledgments. This research was supported by the GACR project No. 14-02424S.

References

1. Atanassov, K.T.: Intuitionistic Fuzzy Sets. Fuzzy Sets and Syst. **20**(1), 87–96 (1986)
2. Anzilli, L., Facchinetti, G., Mastroleo, G.: Evaluation and Ranking of Intuitionistic Fuzzy Quantities. In: Masulli, F., Pasi, G., Yager, R. (eds.) WILF 2013. LNCS, vol. 8256, pp. 139–149. Springer, Heidelberg (2013)
3. Burillo, P., Bustince, H., Mohedano, V.: Some definitions of intuitionistic fuzzy number. First properties. In: Proceedings of the 1st Workshop on Fuzzy Based Expert Systems pp. 53–55 (1994)
4. Çoker, D.: An Introduction to Intuitionistic Fuzzy Topological Spaces. Fuzzy Sets and Syst. **88**(1), 81–89 (1997)
5. Grzegorzewski, P.: Distances and orderings in a family of intuitionistic fuzzy numbers. In: Wagenknecht, M., Hampel, R. (eds.): Proc. of EUSFLAT Conf. 2003, pp. 223–227 (2003)
6. Hung, W.-L., Yang, M.-S.: Similarity Measures of Intuitionistic Fuzzy Sets Based on Hausdorff Distance. Pattern Recognition Lett. **25**, 1603–1611 (2004)
7. Li, D., Cheng, C.: New similarity measures of intuitionistic fuzzy sets and application to pattern recognition. Pattern Recognition Lett. **23**, 221–225 (2002)
8. Mareš, M.: Weak Arithmetics of Fuzzy Numbers. Fuzzy Sets and Syst. **91**(2), 143–153 (1997)
9. Marinov, E.: On Extension Principle for Intuitionistic Fuzzy Sets. Notes on Intuitionistic Fuzzy Sets **20**(3), 34–41 (2014)
10. Stoeva, M.: Intuitionistic Fuzzy Extension Principle. Notes on Intuitionistic Fuzzy Sets **5**(3), 29–30 (1999)
11. Mitchell, H.B.: On the Dengfeng-Chuntian similarity measure and its application to pattern recognition. Pattern Recognition Lett. **24**, 3101–3104 (2003)
12. Xu, Z.S.: Intuitionistic Fuzzy Aggregation Operators. IEEE Transactions on Fuzzy Syst. **15**(6), 1179–1187 (2007)
13. Xu, Z.S., Xia, M.: Induced Generalized Intuitionistic Fuzzy Operators. Knowledge-Based Syst. **24**(2), 197–209 (2011)
14. Xu, Z.S., Yager, R.R.: Some geometric aggregation operators based on intuitionistic fuzzy sets. Int. J. of General Syst. **35**(4), 417–433 (2006)
15. Zadeh, L.A.: Fuzzy Sets. Information and Control **8**(3), 338–353 (1965)

On Fuzzy *c*-Means and Membership Based Clustering

Vicenç Torra(✉)

University of Skövde, Skövde, Sweden
vtorra@his.se

Abstract. Fuzzy *c*-means is one of the most well known fuzzy clustering algorithms. It is usually solved using an iterative algorithm. This algorithm does not ensure that the solution is the global optimum. In this paper we study the distribution of values of the objective function of fuzzy *c*-means.

We also propose a new fuzzy clustering method related to fuzzy *c*-means. The method presumes that the shape of the membership function is known and can be calculated from the cluster centers, which are the only results of the clustering algorihm.

Keywords: Fuzzy *c*-means · Membership based clustering with fuzzy *c*-means membership

1 Introduction

Fuzzy *c*-means [1] and *k*-means are two well known clustering algorithms being the former a fuzzy version of the latter. Both clustering methods are defined in terms of an objective function that has to be minimized.

The most common approach to solve the minimization problem is an iterative process which interleaves a step to compute the membership and another to compute cluster centers. This algorithm converges to an optima. Nevertheless, the algorithm does not ensure that this is a global optimum.

There are applications where an approximate solution is acceptable, and others in which different solutions can be equally acceptable. Nevertheless, there are other applications in which this is a potential problem. This is the case of the application of clustering to evaluate the effectiveneess of data protection methods in data privacy. See [4] for details.

The difficulty of finding the global optimum has been addressed in the literature in several papers. There have been approaches based on e.g. genetic algorithms [2] and shadowed sets and particle swarm optimization (PSO) [5,9].

Alternatively, we can represent the uncertainty associated to the multiple applications of the clustering algorithm obtaining different results. For example, solutions based on interval-valued fuzzy sets have been proposed for this purpose. See e.g. [8].

I. Rojas et al. (Eds.): IWANN 2015, Part I, LNCS 9094, pp. 597–607, 2015.
DOI: 10.1007/978-3-319-19258-1_49

In this paper we study the distribution of values of the objective function for several executions of fuzzy c-means. The experiments show the difficulty of finding a solution with a low value of the objective function for data of small size. Then, we propose a new approach of the fuzzy c-means problem.

The structure of this paper is as follows. In Section 2 we review fuzzy c-means and the most usual algorithm. Then, in Section 3 we study the problem of finding a global solution with this algorithm and how it behaves with some data. We discuss the results. In Section 4 we propose an alternative fuzzy c-means clustering. The paper finishes with some conclusions and lines for future work.

2 Preliminaries

Fuzzy c-means is one of the most used algorithms for fuzzy cluster. Given a data set $X = \{x_1, \ldots, x_n\}$ of n records in a p dimensional space, the goal of the fuzzy clustering algorithm is to find a fuzzy partition with c clusters of this data. The number of clusters c is one of the parameters of the algorithm. A fuzzy partition with c clusters is represented in terms of c membership functions μ_k for $k = 1, \ldots, c$. That is, $\mu_k(x_i) \in [0, 1]$ for $x_i \in X$ and $k \in \{1, \ldots, c\}$. In order that μ defines a fuzzy partition, we need that

$$\sum_{k=1}^{c} \mu_k(x_i) = 1.$$

Then, different clustering algorithms differ on how the membership functions are defined and/or find.

Fuzzy c-means is defined as an optimization problem. That is, given a data set, the solution of fuzzy c-means is the set of membership functions that minimize an expression given a set of constraints.

The function to be minimized is the following one:

$$J_{FCM}(\mu, V) = \{\sum_{k=1}^{c} \sum_{i=1}^{n} (\mu_k(x_i))^m ||x_i - v_k||^2\} \tag{1}$$

Here, v_i for $i \in \{1, \ldots, c\}$ are the centroids, the cluster representatives, of each cluster. This expression roughly means that the distance between a record x_i and the cluster centers assigned (according to the membership) should be minimal.

In Equation 1 we have m which is another parameter of fuzzy c-means. This parameter which should be greater than 1 establishes the degree of fuzziness of the solution. The larger the m, the fuzzier is the solution. When m is near to one, the solution is crisp (and would correspond to the one of k-means). In contrast, when m is large, the algorithm tends to assign memberships equal to $1/c$ to all objects to all clusters. That is, all objects belong to all clusters with the same memberships.

Algorithm 1. Fuzzy c-means

Step 1: Generate initial V

Step 2: Solve $\min_{\mu \in M} J(\mu, V)$ computing:

$$\mu_k(x_i) = \Big(\sum_{j=1}^{c} \Big(\frac{||x_i - v_k||^2}{||x_i - v_j||^2} \Big)^{\frac{1}{m-1}} \Big)^{-1}$$

Step 3: Solve $\min_V J(\mu, V)$ computing:

$$v_k = \frac{\sum_{i=1}^{n} (\mu_k(x_i))^m x_i}{\sum_{i=1}^{n} (\mu_i(x_k))^m}$$

Step 4: If the solution does not converge, go to step 2; otherwise, stop

The constraints of the problem define the set of membership functions μ as a fuzzy partition. They are the following: $\mu_k(x) \in [0, 1]$ for all $x \in X$ and all $k \in \{1, \ldots, c\}$ and $\sum_{k=1}^{c} \mu_k(x) = 1$ for all $x \in X$.

This problem is usually solved using the iterative algorithm described in Algorithm 1. It consists of an iterative process that interleaves two steps. One computes the membership of elements to clusters and the other that computes cluster centers. This corresponds to Steps 2 and 3 in the algorithm. Step 2 is solved as a minimization problem assuming that the cluster centers v_i are known and then optimal memberships μ are computed. Similarly, Step 3 is solved as a minimization problem assuming that the membership values μ_k are known and then optimal cluster centers v_i for these memberships are computed. The expressions to compute μ and v are mathematically obtained from the optimization problems $\min_{\mu \in M} J(\mu, V)$ and $\min_V J(\mu, V)$.

This algorithm leads to a local optimum, but there is no guarantee that the solution is a global optimum. Because of that, as pointed out in the introduction, other approaches have been considered in order to find better or alternative solutions of the optimization problem. That is, the goal is to find a fuzzy partition that is a solution of the minimization problem of Equation 1.

In the next section, we discuss and quantify the difficulty of the classical fuzzy c-means algorithm for finding the optimal solution.

The need to quantify the difficulty of finding a global optimum is partly motivated by our work on data privacy. In our research, as in [4], we need to compare the effectiveness of data protection techniques for some uses on a data set. In the case of clustering, this is done comparing the clustering on the original data set and on the protected data set. Nevertheless, the comparisons are not meaningful when the two applications lead to different local optima.

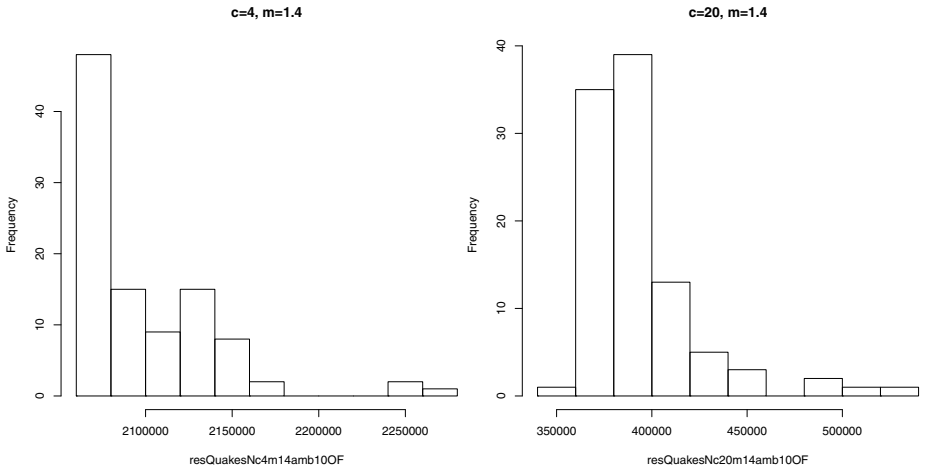


Fig. 1. Histogram of objective functions for 100 executions of FCM with the QUAKES data set. Parameters used: $m = 1.4$ and $c = 4, 20$.

3 Analysis of the Values of the Objective Functions

In order to quantify the difficulty of finding the global optimum of the minimization problem, we have considered the analysis of the values of the objective function on different executions of the fuzzy clustering algorithm.

We have studied the distribution of values of the objective function for a few different data sets and parameters. More specifically, we have considered three data sets IRIS, QUAKES and THEOP. All three datafiles are available in R, and we used the version R supplies.

The IRIS data set consists of 150 records described in terms of 4 numerical variables, the QUAKES data set consists of 1000 records described in terms of 5 numerical variables, and the THEOP file consists of 132 records described in terms of 5 variables. In the case of the THEOP file, the first variable was not used because although codified by numbers it is an ordered variable.

We have clustered them using our own implementation of fuzzy c-means, using different parameters for both m and c. As expected, the larger the c (more clusters) the more difficult to find the global optimum of the objective function. We detail our results below, we do not include figures of the computations for the IRIS data set as they are similar to the other ones.

Figure 2 presents the histogram of the values of the objective function for the THEOP data set. We display the results for $m = 1.4$ and then $c = 4, 10, 20$. The convergence criteria applied in the algorithm is the number of iterations. For each of the parameters we include the histogram using 10 and 20 iterations of the fuzzy c-means algorithm. It can be seen that the results do not change significatively with the number of iterations.

The figures show that only for a small number of clusters the algorithm finds most of the times (more than 80% of the times) an objective function near the minimum. In contrast, for both $c = 10$ and $c = 20$ only 5% of the times the value of the objective function is in the lower class.

Figure 3 shows the histogram of the values for the same THEOP data set with the same c as in Figure 2 but with values of $m = 1.1$ and $m = 1.9$. In this case 10 iterations were used as the convergence criterium. The figures show that with $m = 1.9$ the results are, in general, worse than for $m = 1.1$ and $m = 1.4$.

Figure 1 displays the results for the QUAKES data set. In this case it can be seen that for $c = 4$ about 50% of the times the value of the objective function is in the lowest class. Nevertheless, the algorithm almost always fail to find such a solution for $c = 20$.

Note that although we are considering the class with a lower value of the objective function, it is not necessarily true that the global optimum is in that class. This is so because there is no guarantee that we have already found the global optimum with the 100 executions of fuzzy c-means.

4 An Alternative Formulation for Fuzzy Clustering

The optimization problem that defines fuzzy c-means is based on the assumption that we have two unkown sets of terms. One set consists on the membership functions that define the fuzzy partition, and the second consists on the cluster centroids. Finding a solution that optimizes both sets at the same time is not an easy task (as we have seen in the previous section) and there is no closed-form expression for achieving this.

An alternative way to consider a fuzzy clustering problem is to begin with the assumption that the shape of the membership function is predetermined by the cluster centers and the parameter m. That is, we assume that given v_k for $k \in \{1, \ldots, c\}$ and m we compute μ with a given expression (as e.g. the expression of Step 2 in Algorithm 1) for any object x in the domain. If this is the case, then the objective function of the minimization problem only depends on the values of the centroids. That is, we only need to find the values v.

We formulate a fuzzy clustering algorithm in this way below:

Definition 1. *Let $X = \{x_1, \ldots, x_n\}$ a set of n records in a p dimensional space, a real number $m > 1$, and an integer $c > 1$; then, we define the solution of the membership based clustering with fuzzy c-means membership (MCwFCM) as the minimization of the following objective function*

$$OF = J_{MCwFCM}(v) = \{\sum_{k=1}^{c}\sum_{i=1}^{n}(\mu_k(x_i))^m ||x_i - v_k||^2\} \tag{2}$$

where

$$\mu_k(x) = \Big(\sum_{j=1}^{c}\Big(\frac{||x - v_k||^2}{||x - v_j||^2}\Big)^{\frac{1}{m-1}}\Big)^{-1}. \tag{3}$$

Note that in this problem, we have as input the data X and the parameters m and c; and the output is the set of cluster centers v. The values of the membership values are constructed from the output.

We also want to underline that in this algorithm we have dropped the constraints

- $\mu_k(x) \in [0,1]$ for all $x \in X$ and all $k \in \{1,\ldots,c\}$
- $\sum_{k=1}^{c} \mu_k(x) = 1$ for all $x \in X$.

which we had in the fuzzy c-means algorithm because by construction (Equation 3) the memberships will all be positive and add to one.

Therefore, this is an optimization problem with no constraints. In order to solve it, we need to minimize Equation 2.

It is not possible to find the optimal solution of this problem. Nevertheless, we can compute the gradient of this function which permits us to compute the minimum by the gradient descent. This is done in the remaining part of this section.

Let us consider the objective function, using Equation 3, and rewrite it to distinguish between the v_{k_0} and the other v_k. That is,

$$
\begin{aligned}
OF &= \sum_{k=1}^{c}\sum_{i=1}^{n}\left(\left(\sum_{j=1}^{c}\left(\frac{||x_i - v_k||^2}{||x_i - v_j||^2}\right)^{\frac{1}{m-1}}\right)^{-1}\right)^m ||x_i - v_k||^2 \\
&= \sum_{k=1}^{c}\sum_{i=1}^{n}\left(\sum_{j=1}^{c}\frac{1}{||x_i - v_j||^{\frac{2}{m-1}}}\right)^{-m} ||x_i - v_k||^{\frac{-2m}{m-1}} ||x_i - v_k||^2 \\
&= \sum_{k=1}^{c}\sum_{i=1}^{n}\left(\sum_{j=1}^{c}\frac{1}{||x_i - v_j||^{\frac{2}{m-1}}}\right)^{-m} ||x_i - v_k||^{\frac{-2}{m-1}} \\
&= \sum_{k=1}^{c}\sum_{i=1}^{n}\left(\sum_{j=1,j\neq k_0}^{c}\frac{1}{||x_i - v_j||^{\frac{2}{m-1}}} + \frac{1}{||x_i - v_{k_0}||^{\frac{2}{m-1}}}\right)^{-m} ||x_i - v_k||^{\frac{-2}{m-1}}
\end{aligned}
$$

Let us use D to denote

$$
D = \left(\sum_{j=1,j\neq k_0}^{c}\frac{1}{||x_i - v_j||^{\frac{2}{m-1}}} + \frac{1}{||x_i - v_{k_0}||^{\frac{2}{m-1}}}\right)^{-m}
$$

Then

$$
OF = \sum_{k=1}^{c}\sum_{i=1}^{n} D||x_i - v_k||^{\frac{-2}{m-1}}
$$

Now, let us distinguish in the expression for the objective function the case of $k = k_0$ and the other values of k. We have

$$\begin{aligned} OF &= \sum_{k=1}^{c}\sum_{i=1}^{n} D||x_i - v_k||^{\frac{-2}{m-1}} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c} D||x_i - v_k||^{\frac{-2}{m-1}} \\ &= \sum_{i=1}^{n}\left(\sum_{k=1 k\neq k_0}^{c} D||x_i - v_k||^{\frac{-2}{m-1}} + D||x_i - v_{k_0}||^{\frac{-2}{m-1}}\right) \end{aligned}$$

Let us now consider dOF/dv_{k_0}. From the last expression we find that

$$\begin{aligned} \frac{dOF}{dv_{k_0}} &= \frac{d}{dv_{k_0}}\sum_{i=1}^{n}\left(\sum_{k=1 k\neq k_0}^{c} D||x_i - v_k||^{\frac{-2}{m-1}} + D||x_i - v_{k_0}||^{\frac{-2}{m-1}}\right) \\ &= \sum_{i=1}^{n}\sum_{k=1 k\neq k_0}^{c}\frac{d}{dv_{k_0}}\left(D||x_i - v_k||^{\frac{-2}{m-1}}\right) + \sum_{i=1}^{n}\frac{d}{dv_{k_0}}\left(D||x_i - v_{k_0}||^{\frac{-2}{m-1}}\right) \\ &= \sum_{i=1}^{n}\sum_{k=1 k\neq k_0}^{c}\frac{dD}{dv_{k_0}}||x_i - v_k||^{\frac{-2}{m-1}} + \sum_{i=1}^{n}\left(\frac{dD}{dv_{k_0}}||x_i - v_{k_0}||^{\frac{-2}{m-1}} + D\frac{d||x_i - v_{k_0}||^{\frac{-2}{m-1}}}{dv_{k_0}}\right) \\ &= \sum_{i=1}^{n}\sum_{k=1 k\neq k_0}^{c}\frac{dD}{dv_{k_0}}||x_i - v_k||^{\frac{-2}{m-1}} + \sum_{i=1}^{n}\frac{dD}{dv_{k_0}}||x_i - v_{k_0}||^{\frac{-2}{m-1}} + \sum_{i=1}^{n} D\frac{d||x_i - v_{k_0}||^{\frac{-2}{m-1}}}{dv_{k_0}} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c}\frac{dD}{dv_{k_0}}||x_i - v_k||^{\frac{-2}{m-1}} + \sum_{i=1}^{n} D\frac{d||x_i - v_{k_0}||^{\frac{-2}{m-1}}}{dv_{k_0}} \end{aligned}$$

Now, it is easy to prove that

$$\frac{d||x_i - v_{k_0}||^{\frac{-2}{m-1}}}{dv_{k_0}} = \frac{2}{m-1}||x_i - v_{k_0}||^{\frac{-1-m}{m-1}}$$

We call this last expression Δ. Note that the following expression is also equal to Δ:

$$\frac{d}{dv_{k_0}}\left(\sum_{j=1, j\neq k_0}^{c}\frac{1}{||x_i - v_j||^{\frac{2}{m-1}}} + \frac{1}{||x_i - v_{k_0}||^{\frac{2}{m-1}}}\right) = \Delta$$

Now, using the expressions above, we have that

$$dD/dv_{k_0} = -mD^{-m-1}\Delta.$$

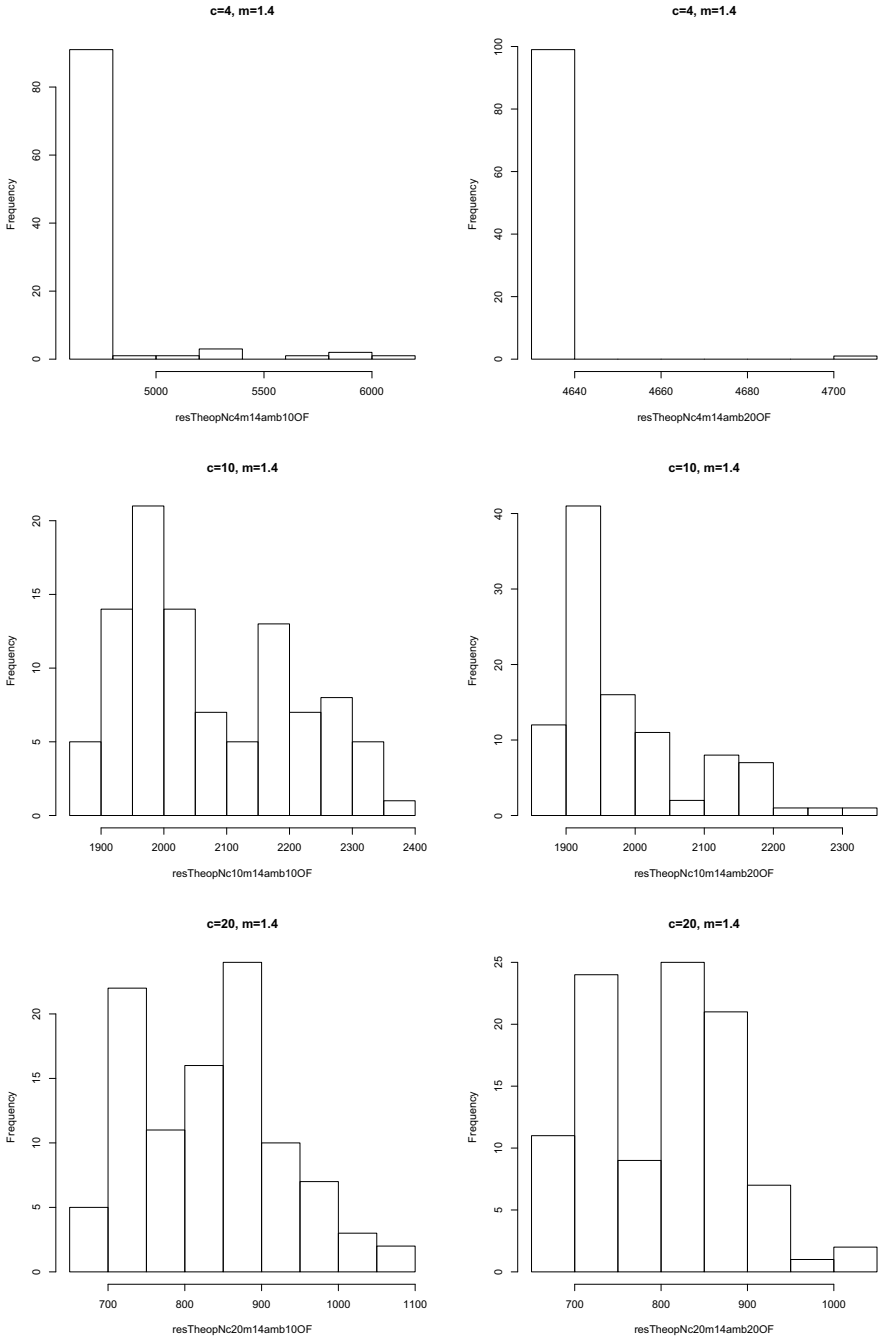


Fig. 2. Histogram of objective functions for 100 executions of FCM with the THEOP data set. Parameters used: $m = 1.4$ and $c = 4, 10, 20$ (from top to bottom). Convergence criteria 10 and 20 iterations (left and right).

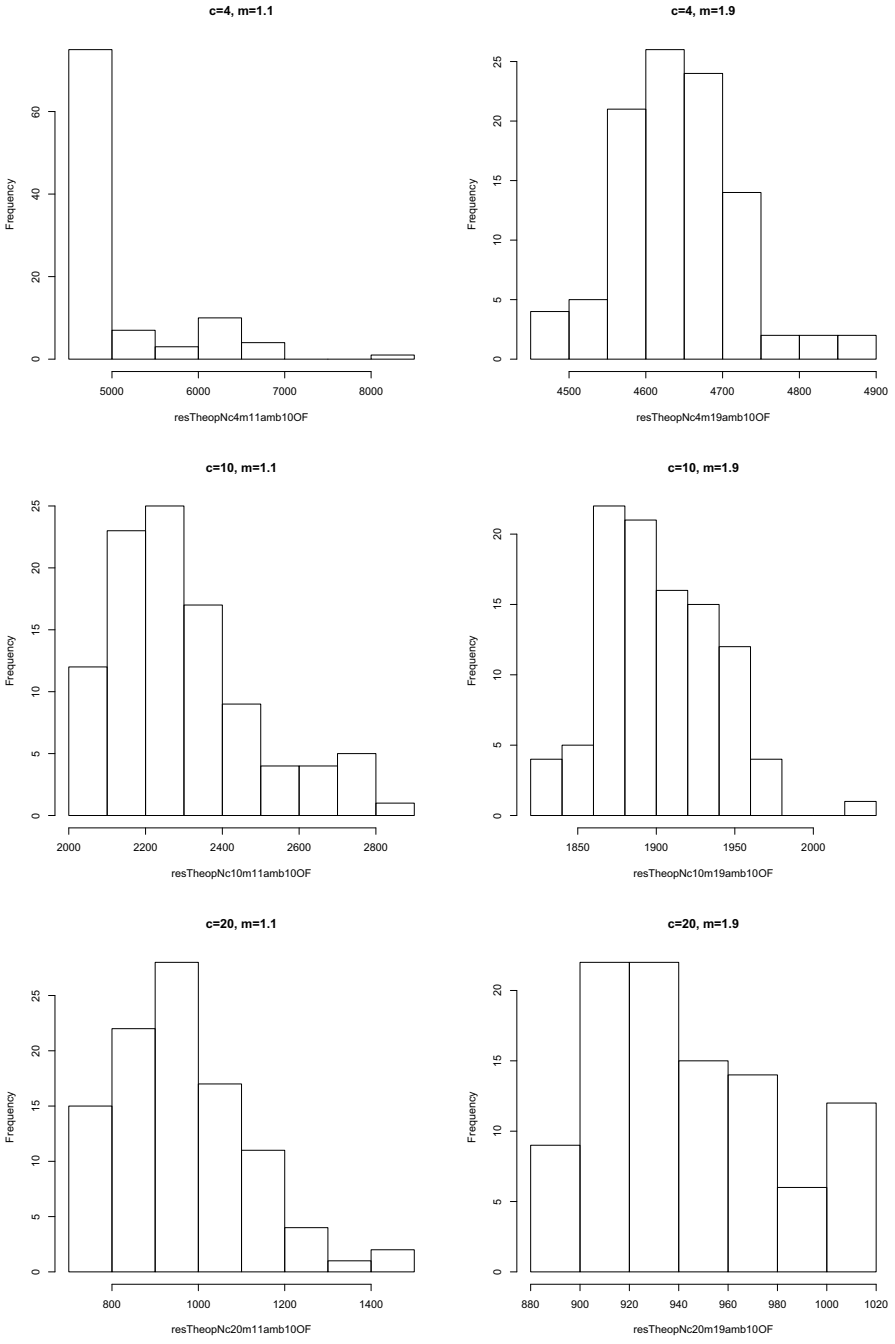


Fig. 3. Histogram of objective functions for 100 executions of FCM with the THEOP data set. Parameters used: $m = 1.1$ and $m = 1.9$ (left and right) and $c = 4, 10, 20$ (from top to bottom). Convergence criteria: 10 iterations.

Therefore,

$$\begin{aligned}\frac{dOF}{dv_{k_0}} &= \sum_{i=1}^{n}\sum_{k=1}^{c}\frac{dD}{dv_{k_0}}||x_i - v_k||^{\frac{-2}{m-1}} + \sum_{i=1}^{n} D\frac{d||x_i - v_{k_0}||^{\frac{-2}{m-1}}}{dv_{k_0}} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c} -mD^{-m-1}\Delta||x_i - v_k||^{\frac{-2}{m-1}} + \sum_{i=1}^{n} D\frac{2}{m-1}||x_i - v_{k_0}||^{\frac{-1-m}{m-1}} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{c} D^{-m-1}\frac{-2m}{m-1}||x_i - v_{k_0}||^{\frac{-1-m}{m-1}}||x_i - v_k||^{\frac{-2}{m-1}} + \sum_{i=1}^{n} D\frac{2}{m-1}||x_i - v_{k_0}||^{\frac{-1-m}{m-1}}\end{aligned}$$

This expression permits us to compute analytically the gradient of the objective function of the proposed fuzzy clustering. Recall that the selection of k_0 is arbitrary. With the gradient we can apply the gradient descent (see e.g. [3]) to find the solution of the membership based clustering with fuzzy c-means membership.

5 Summary and Future Work

In this paper we have studied the behavior of the objective function of fuzzy c-means algorithm. We have considered three different data files and different values for the parameters m and c. We have seen that for relatively small problems, as e.g. the QUAKES data file which only consists of 1000 records and 5 variables, the fuzzy c-means converges for most of the cases to non-optimal solutions. We have also seen that the larger is the file and the larger the dimensions of the file, the more difficult is to find the global optimum.

We have proposed an alternative to fuzzy clustering in which the membership is not one of the values to be determined by the algorithm, but a value that can be computed from the cluster centers which are the solely values calculated from the algorithm.

As future work we consider to study the values of the objective functions for some other fuzzy clustering algorithms. In addition, we consider the application of the proposed method of fuzzy clustering to real data. Finally, we can consider the problem of using other expressions for the calculation of the membership values. In particular, we can use the expressions of the entropy fuzzy c-means.

Acknowledgments. Partial support by the Spanish MEC (project co-privacy – TIN 2011-27076-C03-03) is acknowledged.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
2. Bezdek, J.C., Hathaway, R.J.: Optimization of fuzzy clustering criteria using genetic algorithms. In: Proc. IEEE Conf. on Evolutionary Computation vol. 2, pp. 589–594 (1994)

3. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: Numerical Optimization. Springer (2006)
4. Cano, I., Ladra, S., Torra, V.: Evaluation of Information Loss for Privacy Preserving Data Mining through comparison of Fuzzy Partitions. In: Proc. FUZZ-IEEE 2010/WCCI (ISBN: 978-1-4244-6919-2) (2010)
5. Mehdizadeh, E., Sadi-Nezhad, S., Tavakkoli-Moghaddam, R.: Optimization of fuzzy clustering criteria by a hybrid PSO and fuzzy c-means clustering algorithm. Iranian Journal of Fuzzy Systems **5**(3), 1–14 (2008)
6. Miyamoto, S.: Introduction to fuzzy clustering (in Japanese). In: Morikita, J. (ed.) (1999)
7. Miyamoto, S., Ichihashi, H., Honda, K.: Algorithms for Fuzzy Clustering. Springer (2008)
8. Torra, V., Miyamoto, S.: A definition for I-fuzzy partitions. Soft Computing **15**(2), 363–369 (2011)
9. Zhang, J. Shen, L.: An Improved Fuzzy c-Means Clustering Algorithm Based on Shadowed Sets and PSO. Computational Intelligence and Neuroscience, **2014**, Article ID 368628, 10 pages, (2014). doi:10.1155/2014/368628

Author Index