This volume contains original contributions carefully selected which are derived from PhD students' researches in Pattern Recognition and related areas. These contributions came from the following institutions:

- Instituto Nacional de Astrofísica, Óptica y Electrónica (Puebla, México)
- Universidad de las Américas (Puebla, México)
- Advanced Technologies Application Center (Havana, Cuba)
- University of Informatics Sciences (Havana, Cuba)
- Instituto Tecnológico de Chihuahua (Chihuahua, México)
- Centro Nacional de Investigación y Desarrollo Tecnológico (Cuernavaca, México)

The volume can be useful for both researchers and students interested in Computer Sciences, mainly in recent advances on Pattern Recognition.

ISSN: 1870-4069
www.ipn.mx
www.cic.ipn.mx

SEP

INSTITUTO POLITÉCNICO NACIONAL
"La Técnica al Servicio de la Patria"

Special Issue:
Advances in Pattern Recognition

J. A. Olvera-López
J. F. Martínez-Trinidad
J. A. Carrasco Ochoa
J. Salas Rodríguez
G. Sanniti di Baja
(Eds.)

Vol. 61

RESEARCH IN COMPUTING SCIENCE

ISSN: 1870-4069

Special Issue:
Advances in Pattern Recognition

J. A. Olvera-López
J. F. Martínez-Trinidad
J. A. Carrasco-Ochoa
J. Salas Rodríguez
G. Sanniti di Baja
(Eds.)

Vol. 61

RCS Research in Computing Science

Table of Contents

Índice

Page/Pág.

Advances in Pattern Recognition

Accelerating frequent itemsets mining on data stream: a proposal

Lázaro Bustio-Martínez[1,2], René Cumplido-Parra[2], José Hernández-Palancar[1], and Claudia Feregrino-Uribe[2]

[1] Advanced Technologies Application Center.
7^a ♯ 21812 e/ 218 y 222, Rpto. Siboney, Playa, C.P. 12200, Havana, Cuba.
{lbustio,jpalancar}@cenatav.co.cu
[2] National Institute for Astrophysics, Optics and Electronic.
Luis Enrique Erro No 1, Sta. Ma. Tonantzintla, 72840, Puebla, México.
{rcumplido,cferegrino}@ccc.inaoep.mx

Abstract. Frequent itemsets mining is a widely used technique in Data Mining and has application in many areas of common life: data streams analysis is one of such applications. Several sequential and parallel algorithms have been proposed but traditional approaches are ineffective to face data streams, due to the speed of data in such data source and the impossibility of access them twice or more. In this paper we propose a research driven to obtain parallel algorithms capable to perform frequent itemsets mining on data streams using hardware reconfigurable.

1 Introduction

In Data Mining is very useful to record all the occurrences of certain patterns to be used in forthcoming tasks. One of such tasks is frequent itemsets mining where frequent itemsets are those sets of data items that can be found always together (without concerning the apparition order) more than a given number of occurrences in some data source. In other words, the goal of frequent itemsets mining is to determine which itemsets in a database (or any other data source) commonly appear together.

In early 90's Agrawal proposed an algorithm for frequent itemsets mining named *Apriori*[1]. Apriori was the first approach proposed for this task and also the simplest. The process to find all frequent itemsets are computationally intense. Plenty of algorithms have been created to perform frequent itemsets mining tasks. All of these algorithms requires a lot of powerful computational resources and time to solve the combinatorial explosion of itemsets that can be found in a dataset. When it is processing very large datasets, this issue get worst provoking the task can not be achieved and consequently they fail. This is mainly due to the presence of thousands of different patterns or the use of a too low threshold of support (see *Support* concept in 3).

In addition, in recent times there has been a data explosion and classical approaches (like frequent itemsets mining) to extract hidden knowledge from those huge amounts of data fail while trying to process them. Data generating

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 3-12
Paper Received 20-03-2013 and Accepted 22-04-2013

rate is growing exponentially while data mining applications performance has only increased by 10-15% [10]. This fact shows that the classical paradigm of these algorithms is not suitable.

One scenario that is gaining a lot of attention of researchers is the data streams analysis. A data stream can be seen as unordered and potentially infinite flow of data at high velocity rates. Analizing data streams is an emerging need and it can be found in video and audio streams, network traffic, commercial transactions, etc, but those applications need to be as fast as they can so hardware-based approaches have been proposed. Frequent itemsets mining in such scenario addresses new challenges and only in [4] is conducted a research to frequent itemsets mining on data streams.

This paper is structured as follow: in the next section, different platforms to implement algorithms are explained. Section 3 explains the theoretical basis of the frequent itemset mining. A review of state-of-the-art is addressed in section 4 while section 5 presents the proposed research objectives. This paper is concluded in section 6.

2 Platforms for algorithms implementation

There are two main approaches to implement algorithms in general. The first one consists in building Application Specific Integrated Circuits (ASICs). They are designed and built specifically to perform a given task, and thus they are very fast and efficient. ASICs can not be modified after fabrication process and this is their main disadvantage. If an improvement is needed, the circuit must be re-designed and re-built, with the costs that this entails. The second one consists in using a General Purpose Processor (GPP) which is programmed by software; it executes the set of instructions that are needed by algorithms. Changing the software instructions implies a change in the algorithms behavior. This results in a high flexibility but the performance will be degraded. To accomplish certain function, the GPP, first must read from memory the instructions to be executed and then decode their meaning into native GPP instructions to determine which actions must be done. Translating the original instructions of an algorithm introduces a delay.

FPGAs can be seen as a mesh of basic logic gates interconnected together and its functionality is customizable at runtime. The connections between the logic gates are also configurable. The architecture of a FPGAs is based on a large number of logic blocks which perform basic logic functions. Because of this, an FPGA can implement from a simple logical gate, to a complex mathematical function. FPGAs can be reprogrammed, that is, the circuit can be "erased" and then, a new architecture that implements a brand new algorithm can be implemented. This capability of the FPGAs allow the creation of fully customized architectures, reducing cost and technological risks that are present in traditional circuits design. Fig. 1 represent the position of FPGAs between ASICs and GPP.

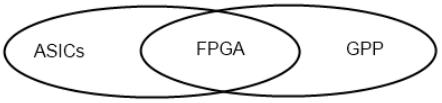


Fig. 1. FPGAs combines the advantages of ASICs and GPP.

3 Theoretical basis

Frequent itemsets mining was introduced by Agrawal et al. back in 1993 [1] and it is used for finding common and potentially interesting patterns in databases. In this scope the data are represented by means of transactions, each of which is a set of items labeled by a unique ID. The purpose of frequent itemsets mining is to find the most frequently-occurring subsets from the transactions. The frequency of the subset is measured by support ratio, which is the number of transactions containing the subset divided by the total number of transactions in the database. Formalizing, let I be a set of items:

Definition 1 (Itemset). *A itemset X is set of items over I such $X = i_i, ..., i_k \subseteq I$. If a set X contains k items then the set X is called* k-itemset. *Normally is considered that the items in an itemset are lexicographically ordered.*

Definition 2 (Transaction). *A transaction over I is a couple $T = (tid, I)$ where tid is the transaction identifier and I is an $X \subseteq I$ itemset. A transaction $T = (tid, I)$ is said to support an itemset $X \subseteq I$, if $X \subseteq I$.*

Definition 3 (Support). *The support of an itemset X in D is the number of transactions in D that supports X:*

$$Support(X, D) = |\{tid | (tid, I) \in D, X \subseteq I\}| \tag{1}$$

An itemset is called $frequent$ if its support is no less than a given absolute minimal support threshold ϕ_{abs}, with $0 \leq \phi_{abs} \leq |D|$. The frequent itemsets discovered does not reflect the impact of any other factor except frequency of the presence or absence of an item.

3.1 Issues in frequent itemsets mining.

The theoretical basis of frequent itemsets mining that has been presented can be applied to both databases and data streams scenarios. The major problem with frequent itemsets mining methods in both scenarios is the explosion of the number of results, so it is difficult to find the most interesting frequent itemsets. Normally must be faced the following disadvantages (databases and data streams scenarios): (a) huge data volumes (order to gigabytes); (b) elevate number of transactions; (c) limited computing resources (like memory) and (d) unpractical processing times.

The main issue in databases (and also in data streams) scenario is concerned to the high number of items to handle (and memory consumption). Let be n the number of single items in database, the number of candidates frequent itemsets is 2^n, or the same, this problem has computational complexity of $O(2^n)$. Handling those huge amount of data is a challenging task, and strategies for efficient data access and data memory maintaining are needed [12].

In other kind of applications, data streams are becoming too attractive and they have been used in many real life applications. Data streams can be defined as a continuous, ordered and potentially infinite sequence of items in real time and also share the same limitations of frequent data streams on databases. Considering the three characteristics of data streams (continuity, expiration and infinity), new limitations are added to frequent itemsets mining problem on data streams: (a) It is impossible to store the stream for latter processing; (b) items in a data stream must be processed just once and (c) items must be processed in a very short time interval.

In other words, frequent itemsets mining on data streams is a particular case of frequent itemsets mining on databases that includes some extra challenges, besides the issues that entails frequent itemsets mining on databases.

4 Algorithms review in hardware

Hardware implementations of algorithms take advantage of inner parallelism of hardware device. In consequence such devices gain every day more attention to be used as development platforms. After a proper review of the state-of-the-art, it can be organized as shown table 1.

Analyzing the revised literature we can realize that frequent itemsets mining on data streams using hardware reconfigurable is an interesting research area that has not been studied in deep. So, we consider that it is worth to propose new parallels algorithms to face such task.

4.1 Frequent itemsets mining based on hardware acceleration

Algorithms for frequent itemsets mining can be divided into two groups: Apriori-based and FP-Growth-based.

Apriori-based. The algorithms that follow the Apriori-based scheme in hardware require loading the candidates itemsets and the database into the hardware. This scheme is limited by the capacity of the chosen hardware platform: if the number of items to handle is larger than the hardware capacity the items must be loaded separately in many consecutive times degrading performance. In consequence, the support counting must be executed several times. Since the time complexity of those steps that need to load candidates itemsets or database items into the hardware is in proportion to the number of candidates itemsets and the number of items in the database, this procedure is very time consuming.

Table 1. Principal algorithms and architectures for the frequent itemsets mining problem on data streams.

Title	**Author**	**Year**	**Based**	**Source**
An Architecture for Efficient Hardware Data Mining Using Reconfigurable Computing Systems. [2]	Baker	2006	Apriori	BD
Hardware Enhanced Mining for Association Rules. [5]	Liu	2006	Apriori	Stream
Hardware-Enhanced Association Rules Mining With Hashing and Pipelining. [11]	Wen	2008	Apriori	BD
Novel Strategies for Hardware Acceleration of Frequent Itemset Mining With the Apriori Algorithm.[10]	Thöni	2009	Apriori	BD
Mining Association Rules with Systolic Trees.[8]	Sun	2008	FP-Growth	BD
A Reconfigurable Platform for Frequent Pattern Mining.[7]	Sun	2008	FP-Growth	BD
A Highly Parallel Algorithm for Frequent Itemset Mining. [6]	Mesa	2010	FP-Growth	BD
Design and Analysis of a Reconfigurable Platform for Frequent Pattern Mining. [9]	Sun	2011	FP-Growth	BD

In addition, numerous candidates itemsets and a huge database may cause a bottleneck in the system.

In the revised literature was found only one paper referred to frequent itemsets mining on data streams scenario that uses hardware reconfigurable. Liu et al. in [5] proposed a hardware-enhanced mining framework and an Apriori-like algorithm to mine frequent temporal patterns[3] from data streams. This architecture is specially designed to mine those itemsets of length 1 and 2, because the computing of L1- and L2-itemsets is the most time-consuming task in their algorithm, so they proposes offload this operation to hardware to enhance the global performance. They states in their work the main issues when dealing with data streams, which are consistent with that was explained in section 3.1. According to the authors, the novelty in this hardware-enhanced approach resides in the transformation of the items transactions from a data streams into a matrix structure and efficiently map operations for discovering frequent items to highly efficient hardware processing unit. A "receiving-storing-processing" approach is used to perform the transformation of transactions of stream into matrix. Experiments on synthetic data set shown that the throughput is two order of magnitudes larger than its software counterpart does.

[3] Frequent Temporal Pattern is referred to those items or itemset that are frequent in some time period.

FP-Growth-based. FP-Growth is one of the fastest and efficient algorithm in the frequent itemsets mining state-of-the-art, which has been implemented in several ways, including sequential and parallel implementations. FP-Growth is based on a prefix tree representation of the given database of transactions (called an FP-tree). The FP-tree representation allow saving considerable amount of memory for storing the transactions. In FP-tree representation every transaction are stored as a string in the tree along with its frequency. Fig. 3 shows the FP-tree presentation for transactions shown in Fig. 2.

ID	Items
1	B, C, D
2	B, C
3	A, C, D
4	A, C, D
5	A, B, C
6	A, B, C
7	A, B, D

Fig. 2. Transaction BD

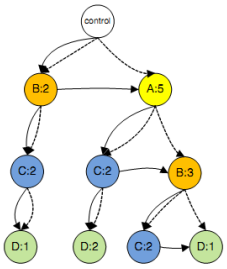


Fig. 3. FP-tree representation of BD in Fig. 2

In 2008 Song Sun and Joseph Zambreno proposed an architecture [8] to speed up the association rules mining process based on FP-Growth. To emulate the FP-Tree data structure they proposed a new hardware structure named *systolic tree*[4]. A systolic tree could be seen as a tree where each node is a processing unit which has their own logic. Fig. 4 shows the systolic tree built for the FP-tree represented in Fig. 2.

The main idea implemented in this architecture is to build a lexicographic tree while items flows through the systolic tree. When all the database is mapped into the systolic tree, each PE will contain the frequency of the current item.

Also in 2008, Sun and Zambreno propose a new hardware architecture for frequent itemsets mining using a systolic tree [7]. As similar with [8] the goal of this architecture is to emulate the original FP-Growth algorithm while achieving a much higher throughput. The main apport in this paper is that Sung et al. modifies the original scheme introduced in [8] by eliminating the counting nodes (see Fig. 2), and providing a new count mode algorithm.

In 2010 Mesa et al. [6] proposed a novel architecture that used FP-Growth as starting point. They proposed a vertical bit vector data layout to represent items and transactions. This layout allows to calculate the support by using logical AND and OR operations. Mesa et al. define a two-dimensional matrix to

[4] In VLSI terminology, a systolic tree is an arrangement of pipelined processing elements in a multi-dimensional tree pattern.

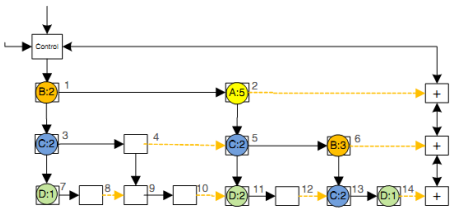


Fig. 4. Systolic tree architecture that emulates the FP-tree data structure represented in Fig. 2.

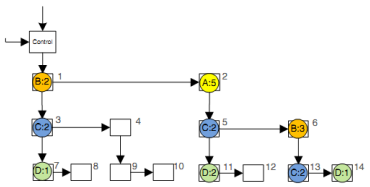


Fig. 5. Enhanced systolic tree architecture that emulates the FP-tree data structure represented in Fig. 2. Notice that the Counting PE was eliminated in this approach.

store the database where the columns represent the elements of the dataset and the rows represent the transactions. Using such data representation, Mesa et al. proposed an algorithm that is based on a search over the solution space through the equivalence class considering a lexicographical order over the items. This is a two-dimensional search (bottom-down, right-left), both breath and depth is performed concurrently. The algorithm proposed do not need a candidate generation stage and it uses a binary tree structure of processing elements. This structure represent a systolic tree and it was chosen because it allows to exponentially increase the concurrent operations at each processing step. For the chosen device, only 11 items can be processed. Experiments demonstrates that the proposed architecture outperforms the architecture reported in [7] almost in one order of magnitude. Also, the architecture performs better when the density of the database and the number of frequent itemsets increases.

Once again, in 2011 Sun et al. went back to [7] and explain in a more detailed manner the systolic tree architecture and their approach [9]. In this paper the authors expose the same idea reported in [7], but they extend their paper with more detailed explanations about systolic tree and the working modes of their architecture. Also, new experiments were performed demonstrating that the systolic tree architecture can outperform the best known FP-Growth implementation [3]. Experiments demonstrate that systolic tree is a valid architecture to mine frequent patterns. For those dataset having sizes that can be placed directly in the device, the systolic tree architecture always outperform FP-Growth. In such cases that the dataset could not be placed directly in the FPGA, a dataset projection must be used and chosen. In this work, Sun et al. proposes one projection dataset strategy and prove its feasibility. In such cases, FP-Growth outperform systolic tree structure. This behavior is caused by the overhead introduced by the projection strategy for database transaction mapping over FPGA. If the overhead is not amortized by the runtime reduction, the systolic tree algorithm is slower than the original FP-Growth algorithm.

In this research proposal, we will focus on FP-Growth-oriented and data-stream-oriented algorithms because of this approach is closer to hardware nature and in the mining process, items are accessed only once.

4.2 Research Problem

Modern applications generate huge data volumes in a data streams way. Due to the increase of this kind of applications it is necessary obtain useful knowledge from those data streams. As it was previously defined, a data streams are a continuous, ordered and potentially infinite sequence of items in real time where data arrives without interruptions at a high speed. Also, data can be accessed only once and the only assumption that we can make about bounds of streams is that the total number of data is unbounded. It is unrealistic to store all items of data streams to process them offline. These characteristics impose an extra difficulties to algorithms and systems that process such data sources.

Due to the high incoming rate, the impossibility to store the data and the huge volumes of items in streams, softwares that analyze they can not process exhaustively all items. The supporting hardware and software are not capable to deal with such intense processing. Instead, softwares use an "approximate" processing approach. That is, they do not analyzes all and each one of items that are present in a flow; instead they use some heuristic or probabilistic approach to determine which item is the most likely to contain the desired information. There are some applications where this approach is not valid, for example in an intrusion detection system or network analysis system. In these kind of applications, the exhaustive data analysis and realtime response are extremely valuable. To provide an realtime response and an exhaustive item processing is needed to use a more specific hardware such as FPGAs. FPGAs can perform tasks in a high parallel fashion and this allow to process all items in a data streams exhaustively in realtime.

Frequent itemsets mining is one technique that is very used in data knowledge extraction and have been used with success in data streams scenario. The frequent itemsets mining on data streams using software is performed in an approximate way. To mine frequent itemsets on data streams exhaustively and responding in realtime, is needed to develop new parallel algorithms and the use of custom hardware architectures. In the reviewed literature there is only one architecture to mine frequent itemsets on data streams [4].

Summarizing, data streams are a modern data source that are gaining interest in recent applications. Traditional approaches are not suitable to be used in data streams and they introduce new challenges to frequent itemsets mining. Due to the growing number of applications that produces data streams and the impossibility to process them in a proper manner, it is worth to propose algorithms that can analyze data streams in real time. To pursuit such goal, we propose to use FPGAs as development platform taking advantage of the parallel capabilities of FPGAs.

4.3 Aims

The general aim of this research work is :

To develop new methods for frequent itemsets mining on data streams that outperform the state-of-art algorithms in data streams analysis in a more efficient and effective manner.

The specific aims proposed are:

1. To select an optimal method to perform the transactions separation in a data streams.
2. To obtain a new method for items representation that can be used in frequent itemsets mining on data streams.
3. To develop new algorithms for frequent itemsets mining that use the separation method and items representation proposed earlier that can outperform the state-of-art algorithms in a more efficient and effective manner.
4. To obtain a parallel hardware implementation of the specific objectives mentioned above that can perform frequent itemsets mining at least 10 times faster (without compromising efficiency) than state-of-the-art software implantations, and 2 times faster than state-of-the-art hardware implementations.

4.4 Contributions

The contributions expected of this proposal are:

1. A new compact data representation that can be used in data stream handling.
2. A design of a parallel one-pass algorithm to mine frequent itemsets on data streams that uses the compact data representation presented before.
3. A parallel implementation in hardware (specifically in FPGAs) of the proposed algorithm.
4. An analysis of the performance of the proposed algorithm and an analysis of the tradeoff between parallelism-performance that allows to estimate the scalability of the implemented system.

5 Conclusions

Frequent itemsets mining is a widely used technique in Data Mining applications. Hence, many researchers are currently engaged in proposing methods for improve and accelerate such process while the data sources increase their sizes. In recent times the streams of data have gained more interest due to its applications, and frequent itemsets mining are being used to obtain new knowledge from those unorganized and continuous data flows.

Classical methods are ineffective when they deal with data streams so it is necessary to explore other alternatives methods such us parallel algorithms that take fully advantages over hardware devices like FPGAs. Data flows can be seen

as an especial case of data bases except that data flows introduce some restrictions. So, the aim of this work is to develop new parallel algorithms than can perform the frequent itemsets mining over data streams using reconfigurable hardware that outperform reported reported results for both software and hardware approaches.

References

1. Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.
2. Zachary K. Baker and Viktor K. Prasanna. An architecture for efficient hardware data mining using reconfigurable computing systems. In *Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, FCCM '06. IEEE Computer Society.
3. Christian Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, OSDM '05, pages 1–5. ACM, 2005.
4. Chih hsiang Lin, Ding ying Chiu, and Yi hung Wu. Mining frequent itemsets from data streams with a time-sensitive sliding window. In *In SDM*, 2005.
5. Wei-Chuan Liu, Ken-Hao Liu, and Ming-Syan Chen. Hardware enhanced mining for association rules. In *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, pages 729–738. Springer-Verlag, 2006.
6. Alejandro Mesa, Claudia Feregrino-Uribe, Ren Cumplido, and Jos Hernndez-Palancar. A highly parallel algorithm for frequent itemset mining. In J.F. Martnez-Trinidad, J.A. Carrasco-Ochoa, and J. Kittler, editors, *Advances in Pattern Recognition*, volume 6256 of *Lecture Notes in Computer Science*, pages 291–300. Springer Berlin Heidelberg, 2010.
7. Song Sun, Michael Steffen, and Joseph Zambreno. A reconfigurable platform for frequent pattern mining. In *Proceedings of the 2008 International Conference on Reconfigurable Computing and FPGAs*, RECONFIG '08, pages 55–60. IEEE Computer Society, 2008.
8. Song Sun and Joseph Zambreno. Mining association rules with systolic trees. In *FPL*, pages 143–148. IEEE, 2008.
9. Song Sun and Joseph Zambreno. Design and analysis of a reconfigurable platform for frequent pattern mining. *IEEE Transactions on Parallel and Distributed Systems*, 22:1497–1505, 2011.
10. David W. Thöni and Alfred Strey. Novel strategies for hardware acceleration of frequent itemset mining with the apriori algorithm. In *19th International Conference on Field Programmable Logic and Applications, FPL 2009*, pages 489–492. IEEE, 2009.
11. Ying-Hsiang Wen, Jen-Wei Huang, and Ming-Syan Chen. Hardware-enhanced association rule mining with hashing and pipelining. *IEEE Trans. on Knowl. and Data Eng.*, 20(6):784–795, June 2008.
12. Guizhen Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 344–353. ACM, 2004.

Temporal self-organized meta-learning for predicting chaotic time series

Rigoberto Fonseca[1], Pilar Gómez-Gil[1]

[1] National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Mexico
rfonseca@inaoep.mx, pgomez@acm.org

Abstract. To predict future values in chaotic systems is difficult but indispensable in several real applications. Over the last years, some authors have been focusing on a meta-learning process of how to combine models to improve prediction accuracy. This research proposal addresses the meta-learning problem of how to combine models using different parts of the prediction horizon. Our aim is to improve the long-term prediction achieved by the current state of the art. We propose to split the prediction horizon in three parts: short, medium and long term prediction horizons. Next in each horizon, we can extract knowledge about what model has the best performance. Thus, we can improve the long-term prediction using different models in each prediction horizon. However, the search space increases and poses nontrivial difficulties because the models could be combined in many ineffective ways. To avoid that, we propose the use of auto-organization. In this paper, we present some preliminary results of our first idea, combining models in different prediction horizons.

Keywords: meta-learning, time series prediction, chaotic time series, self-organization.

1 Introduction

Chaotic time series are cataloged as unpredictable, due its high sensibility to initial conditions [1]. Despite of that, many applications deal with chaotic systems and require a reasonable estimation of future values. For this reason, many domains are looking for an improvement of the accuracy obtained by current prediction models, for example in financial applications, load forecasting or wind speed [2]. Nevertheless, the problem of predicting multi-step-ahead, based on data captured from the chaotic system, is still an open problem [2]. Several works have tackled this problem mainly using statistical models and models based on computational intelligence. Available forecasting algorithms can be roughly divided into a few groups. Examples of simple algorithms are moving average and single exponential smoothing. Complex systems, commonly used by statisticians, are based on ARIMA models. Examples of models based on computational intelligence include neural networks and support vector machines. In addition, models have been used stand-alone or as a combination of several strategies [3].

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 13-22
Paper Received 20-02-2013 and Accepted 22-04-2013

In an effort to find the best predictors, Crone et al. [4] analyzed the results obtained by different models competing in the international forecasting tournament NN3 [5]. From that analysis, they concluded three important ideas: combinations of models obtained the best results; some models have better performance than other models depending on the number of steps to predict, that is, the size of the prediction horizon; data features determine the relative performance of different models.

A very important problem when using a combination strategy is to decide what models must be combined and how to combine them. The process used by human experts starts with inspecting the data. Next, the models are selected and adjusted according to their experience. High time and money costs of expert's analysis motivate finding automatic approaches. In the last years, several works have been published related to this issue. For example, Lemke and Gabris [6] presented an interesting work using meta-learning. Meta-learning automatically induces a meta-model from a meta-training set, (data about training data). Given a new prediction task, this meta-model is able to return the best model or combination of models chosen from a model set [7]. The authors extracted features from around 222 time series. This collected data is the meta-data used to train an expert system. The authors outperformed individual methods and combinations of all methods involved in their experiments. Other researchers have obtained good results using self-organization for building combinations of classifiers on no time-dependent domains, for example [8] [9]. Inspired by these successes, we want to investigate how predictor models can cooperate in a self-organized way. Besides, we want to include the use of different prediction horizons in the meta-learning process.

This paper is organized as follows: Section 2 describes the involved problem, including main concepts associated with this research, research questions, objectives and main contributions; section 3 describes the proposed methodology to achieve the objectives. As a starting point, we empirically evaluated if a combination of models in different prediction horizons could improve the prediction accuracy. We call this strategy "temporal combination," described in section 4. Section 5 presents an experiment comparing temporal combination with the most used combination strategy, known as average of predictions. Finally, section 6 presents some conclusions.

2 Problem Statement

2.1 Main concepts

For a time-series prediction system, a sequence of n elements sampled from the past forms a training series; the sequence of m values to predict is known as a prediction horizon. The first future value to be estimated is represented by x_{n+1}. If the estimation of this value is calculated using d past values, a model F that returns a future value may be described as:

$$x_{n+1} = F(x_n, x_{n-1}, \dots, x_{n-d}) \tag{1}$$

A critical factor in predicting time series is to determine the value of d. Chaos theory contains some interesting ideas for finding suitable values for this regard.

The sequence $\{x_{n+1}, x_{n+2}, \dots, x_{n+m}\}$ represents a prediction horizon greater than one, known as multi-step ahead prediction. There are two forms to archive this sequence; one is estimating the complete horizon in a single iteration. A second strategy, known as iterative prediction [4] and used in this research, consists of estimating one value each time, using the previous predicted value for calculating the next prediction.

For many prediction applications, the best results have been achieved combining different models [5]. Diversity among the members of a set of models is deemed to be a key issue in models combination [10]. There are many strategies for combining models. The simple average of predictions is one of the most used, due to its simplicity and good accuracy. In this strategy, each prediction element is the average of all model's estimations. Let C be the prediction horizon, obtained as the average of predictions of k models. Then the elements of C are:

$$c_i = \frac{1}{k}\sum_{j=1}^{k} x_i^j \tag{2}$$

where x_i^j is the i-prediction, $0 \leq i \leq m$, obtained from the jth-model.

In general, the accuracy of a model comes from comparing their estimation $\hat{x}$ with the corresponding real values over the prediction horizon. There are several metrics for this error estimation, being the most used the mean square error (MSE) and the symmetric mean absolute percentage error (SMAPE) [11], defined as:

$$MSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{x}_i - x_i)^2} \tag{3}$$

$$SMAPE = \frac{1}{m}\sum_{i=1}^{m}\frac{|\hat{x}_i - x_i|}{\frac{1}{2}(\hat{x}_i + x_i)}100 \tag{4}$$

Meta-learning has become an important tool for designing prediction applications. Castiello and Fenalli [7] state the following definition of meta-learning:

Let A be a set of learning algorithms and T a set of tasks. Let $a_A(t)$ be the best algorithm in A applicable to a specific task t, for each $t \in T$, and $c(t)$ a characterization of the chosen task t. Then a meta-learning process is an automatic mechanism that starting from the meta-data set:

$$\{\langle c(t), a_A(t)\rangle : t \in T\} \tag{5}$$

induces a meta-model which is able to predict, for a new task, the best model in A. Consequently, the construction of meta-data set is a crucial part in the process of meta-learning. The selected features should cluster the time series correctly. That is, to group the most similar and separates the most different.

A system is self-organized if it acquires a temporary or functional spatial structure without specific interference from outside [12]. A good sample of the flexibility and success of self-organization are the Self Organizing Maps (SOM), proposed by Teuvo Khonen [13]. They are a kind of neural network with unsupervised learning. The

training of the SOM uses competitive learning, which starts looking for what is the neuron most similar to the example shown, that is, the winner. Then it uses a collaborative strategy for updating weights of neurons in the winner's neighborhood.

2.2 Research questions

We propose to search for answers to the following questions:

1. How can we automatically find the right methods to combine and the right way to combine them, in order to improve multi-step ahead prediction in a chaotic time series?
2. How can we extract and exploit the knowledge of the models that work best in different prediction horizons?
3. How can self-organization of prediction methods improve the prediction of a combination of prediction methods?

2.3 Objectives

With the aim of answering the research questions, we have the following main objective in this research: *to develop a meta-learning algorithm capable of building, in a self-organized way, combinations of models considering different prediction horizons on chaotic time series.*

The commitment of this research is to obtain a better prediction accuracy than the models presented in the state of the art. We will compare our method mainly with the work of Lemke and Gabris [6] for their good results. Our results are expected to be a statistically significant improvement in prediction accuracy.

To achieve our general objective, we have the following specific objectives:

1. Define general guidelines for combining models in different prediction horizons, in order to improve the multi-step prediction performance.
2. Develop a meta-learning method considering different prediction horizons, to train an expert system builder of combinations of models.
3. Develop a strategy for self-organizing models, promoting collaboration among them during the meta-learning process.

The expected contributions of this research are:

1. A new strategy to combine prediction models, considering different prediction horizons.
2. A time series meta-data builder, able to find the best model in a search space previously defined.
3. A meta-learning method for training an expert system combining models.
4. A self-organized method for meta-learning in the context of predicting chaotic time series.

3 Proposed methodology

Based in the KDD process [14], we defined the main steps for achieving each of the objectives proposed in this research, which are detailed next. Tasks contributing to the development of the method are validated before declaring the task as ended. The complete method will be validated by comparing it with Lemke and Gabris work [6] and other state of the art works.

1. Create a target data set: select a set of chaotic time series and a set of prediction models. Based on the state of the art we have selected the following: statistics models (ARIMA, Random Walk and Exponential Smoothing) and computational intelligence models (Recurrent Neural Networks and Support Vector Machines). This step also includes:
   (a) Data cleaning and preprocessing: remove noise mainly outliers and approximate missing values.
   (b) Data reduction and projection:
      (i) define representative features of time series (e.g. standard deviation, trend, skewness and largest Lyapunov exponent [6]),
      (ii) for each model, define its parameters and possible values (e.g. number of delay neurons, number of neurons in the hidden layer and training algorithm),
      (iii) define a set of basic strategies for combining models (e.g. simple average, stacking with probability distribution [15], and rotation forest [16]).
   (c) Model evaluation: define metrics for evaluating models in the multi-step prediction task. The most common metrics for assessing prediction are MSE and SMAPE (see equations 3 and 4). In addition, run tests of statistical significance, as the null-hypothesis significance test.
   (d) Develop a time series meta-data builder, able to find the best model in a search space previously defined. An interesting option to select the best model can be Monte Carlo cross-validation [17]. Also, review other alternatives for selecting models.
2. Define a strategy for combining models in different prediction horizons.
   (a) Evaluate the existing strategies of combination of models. Decide which of them, if any, allows models to effectively combine and exploit the best performance of different algorithms in different prediction horizons.
   (b) Propose a combination strategy using different prediction horizons.
   (c) Evaluate the proposed strategy comparing it with the best strategies of combination of models found in the state-of-the-art work.
3. Design a meta-learning algorithm considering different prediction horizons, to train an expert system builder of combinations of models.
   (a) Find a meta-learning strategy able to build an expert system to define combinations of models.
   (b) Extend this meta-learning strategy to allow the combination of models in different prediction horizons. A possible combination could be using the short-term model outputs to enhance the initial conditions of a long-term model.

(c) Compare the performance accuracy of expert systems, trained by the two meta-learning strategies, both the original and the extended.

4. Develop a strategy for self-organizing models, to promote collaboration among them during the meta-learning process
   (a) Analyze the current strategies for self-organization literature, particularly those focused on building combinations of models. Include an analysis of negative correlation [18], aimed to seek a diversity of models.
   (b) Extend the meta-learning algorithm obtained from the third objective, adding the selected strategy of self-organization.
   (c) Compare the new meta-learning algorithm with that obtained by the third objective.
5. Develop a prediction system to exploit the ability of the expert system trained.

The following section shows the progress made so far, with respect to the first two points of the proposed methodology.

4 Temporal combination of models

A desired prediction horizon can be divided into three parts, each with the same number of elements, named short-term, medium-term and long-term. Our goal is to combine models with the best performance in short, medium and long term, as illustrated in figure 1. The prediction models are previously trained, and each model predicts the entire horizon. The prediction of the combination will consist of the prediction of the three models in their prediction horizons. The prediction horizons include the left bound but not the right bound, except the long-term prediction that includes the right boundary.

The selection of the best models in each forecast horizon requires some preprocessing. The original training set is divided into two series, one to train the models and other to evaluate the three prediction horizons. For each model, a SMAPE is calculated. The model selected for short-term horizon will be the one with the smallest value of short-term SMAPE averaged for all series. A similar procedure is followed to select models of the horizons of medium and long term.

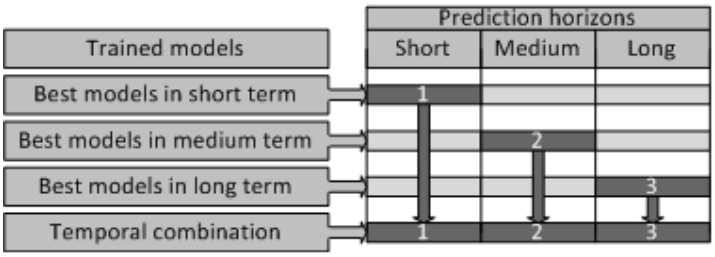


Fig. 1. The temporal combination of prediction models previously trained. The result is composed by the models predictions in their different prediction horizons.

Having defined the models of temporal combination, these are trained with the original training series. Each model produces the full horizon of prediction of size m; predictions are represented by $\{x^s\}$, $\{x^m\}$, and $\{x^l\}$ for selected models of short, medium and long term, respectively. We take a segment from each model prediction. The horizons short, medium and long terms are the same size b. Complete temporal prediction C_T is obtained by joining the three prediction horizons. The binding is expressed in equation 6.

$$C_T = \begin{Bmatrix} \{x_i^s, n+1 \leq i < n+b\} \cup \\ \{x_i^m, n+b \leq i < n+2b\} \cup \\ \{x_i^l, n+2b \leq i < n+3b\} \end{Bmatrix} \tag{6}$$

The prediction C_T is assessed by calculating SMAPE according to short, medium and long term prediction horizons.

5 Preliminary results

In this section, we show some preliminary results. In this experiment, a set of time series was modeled using several prediction models, all based on a NAR neural networks [19]. Then a temporal combination of models was built and compared with two other kinds of combinations.

5.1 Data description

For this first experiment time series were obtained from the NN3 prediction tournament which can be downloaded from: http://www.neural-forecasting-competition.com/NN3/datasets.htm. We used the available reduced set, which consists of 11 time series representing a homogeneous population of empirical business time series. Each training sequence contains between 116 and 126 items, while the prediction horizon is composed of 18 future values for all series. The set of values to predict is called the test set.

The set of models used in this experiment is composed of different non-linear autoregressive neural networks (NAR) [19]. NAR is a recurrent dynamic network with feedback connections enclosing several layers. In this experiment, different models with the same base form are NARs trained with different parameters. Notice that, if a NAR is trained using different algorithms, their weight values will be different and consequently its performance may vary. For that reason, this experiment considers the training algorithm as a parameter.

The parameters used to generate the models are three: the number of delay neurons (1 to 5), the number of neurons in the hidden layer (1 to 5) and the training algorithm (12 in the neural networks toolbox of MATLAB). In total, there are 300 models with the same NAR form. The training algorithms used in this experiment are: 'trainbfg' (BFGS quasi-Newton backpropagation (BP)) 'trainbr' (Bayesian regulation BP), 'traincgb' (Conjugate gradient BP with Powell-Beale restarts), 'traincgf' (Conjugate

gradient BP with Fletcher-Reeves updates), 'traincgp' (Conjugate gradient BP with Polak-Ribiére updates), 'traingd' (Gradient descent BP), 'traingda' (Gradient descent with adaptive learning rate BP), 'traingdm' (Gradient descent with momentum BP), 'traingdx' (Gradient descent with momentum and adaptive learning rate BP), 'trainlm' (Levenberg-Marquardt BP), 'trainoss' (One-step secant BP), 'trainrp' (Resilient BP).

5.2 Experiment setup and analysis of results

The aim of our first experiment is to test whether a temporary combination can perform better than the most commonly used combination in the state of the art. This last, described in section 2.1, is based on averaging the predictions of the all models, throughout the prediction horizon. We compare the results of this temporal combination with a combination made with the three models that best predicted the complete horizon. All models were trained using the same set of training series. Then each model predicted the entire horizon and SMAPE was calculated for each series. Next, we calculated the mean SMAPE of all the series and the top three models with minimum mean SMAPE are selected. Each combination models are trained with all the training set. The outputs of each combination of models are compared with the test set. The experiments were executed 10 times to remove the bias caused by the instability of neural networks. Next, we conducted an evaluation of statistical significance in predicting each series and all predictions. The estimated error is calculated using SMAPE both in the whole prediction horizon as in short, medium and long term horizons. The results are shown in Table 1. First column indicates the ID of the series, the second column shows the error of the combination based on average, the third column shows the error of the combination of the top three models and the fourth column is the error of the temporal combination.

Table 1. Comparison of the three combinations: average, "top three" and temporal in the series of prediction tournament NN3.

No.	Mean SMAPE of Average combination	Mean SMAPE of Top three combination	Mean SMAPE of Temporal combination
1	3.87	4.39	4.34
2	39.05	40.22	57.37
3	97.17	92.41	111.96
4	28.60	28.11	27.56
5	3.10	3.62	3.75
6	4.52	4.52	5.15
7	5.86	5.36	6.89
8	24.93	29.86	29.38
9	11.76	12.68	12.43
10	41.04	32.34	39.28
11	24.92	22.05	23.89
Mean	25.89 +/-26.21	25.05 +/-24.70	29.27 +/-30.75

The last row in the table shows the mean of SMAPE with its corresponding standard deviation. The best result was obtained by the combination of the top three models. However, the statistical significance test showed that the difference between the means of the models is not significant. Indeed, our combinations of models obtained a better performance for some series. The temporal combination obtained the best results in series number 1 and number 4.

Notice that the performance of a model depends on the involved time series. For some cases, the temporal combinations obtained the best result. We expect that increasing the diversity of base models will improve the results of the temporal combination, this according to [10]. On the other hand, the obtained results motivate us to explore different strategies for combining models. Finally, an improvement in the model selection criterion could reduce variability of results.

6 Conclusions

In this paper, we present the initial ideas for the creation of a new algorithm to predict chaotic time series using strategies taken from self-organization, meta-learning and combination of models. From a first experiment, we obtained empirical evidence of the viability of our proposal. This experiment compared the proposed temporal combination with the combination of models based on average strategy, which is the most commonly used in the state of art; also we compared with the combination of the best three models. Since the combination of models in different prediction horizons outperformed the other two strategies, we conclude that it is feasible to design automatic methods able to create temporal combinations. Nevertheless, it is necessary to explore other strategies for combining models, selecting models and extend the set of base models.

Acknowledgements

R. Fonseca thanks the National Council of Science and Technology (CONACYT), México, for a scholarship granted to him, No. 234540. This research has been partially supported by CONACYT, project grant No. CB-2010-155250

References

1. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press (2003)
2. De-Gooijer, J., Hyndman, R.: 25 years of time series forecasting. International Journal of Forecasting 22(3), 443-473 (2006) Twenty five years of forecasting.
3. Makridakis, S., Hibon, M.: The M3-Competition: results, conclusions and implications. International Journal of Forecasting 16(4), 451-476 (2000) The M3- Competition.
4. Crone, S., Hibon, M., Nikolopoulos, K.: Advances in forecasting with neural networks?

Empirical evidence from the NN3 competition on time series prediction. International Journal of Forecasting 27(3), 635-660 (2011)

5. Crone, S., Nikolopoulos, K., Hibon, M.: Automatic Modelling and Forecasting with Artificial Neural Networks– A forecasting competition evaluation. Final Report for the IIF/SAS Grant 2005/6, International Institute of Forecasters (April 2008)
6. Lemke, C., Gabrys, B.: Meta-learning for time series forecasting and forecast combination. Neurocomputing 73(10-12), 2006-2016 (2010)
7. Castiello, C., Fanelli, A.: Computational Intelligence for Meta-Learning: A Promising Avenue of Research. In Jankowski, N., Duch, W., Grabczewski, K., eds. : Meta-Learning in Computational Intelligence 358. Springer Berlin Heidelberg (2011) 157-177
8. García-Pedrajas, N., Hervás-Martínez, C., Ortiz-Boyer, D.: Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification. IEEE Transactions on Evolutionary Computation, 271-302 (2005)
9. Kordik, P., Cerny, J.: Self-organization of Supervised Models. In Jankowski, N., Duch, W., Grabczewski, K., eds. : Meta-Learning in Computational Intelligence 358. Springer Berlin Heidelberg (2011) 179-223
10. Kuncheva, L., Whitaker, C.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. Machine Learning 51, 181-207 (2003)
11. Armstrong, J.: Long-range forecasting from crystall ball to computer 2nd edn. John Wiley & Sons (1985)
12. Haken, H.: Information and Self-Organization: A Macroscopic Approach to Complex Systems 3rd edn. 40. Springer, Stuttgart, Germany (2006)
13. Haykin, S.: Neural Networks A Comprehensive Foundation Second Edition edn. Pearson Prentice Hall (1999)
14. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine 17(3), 37-54 (1996)
15. Dzeroski, S., Zenko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One? Machine Learning 54, 255-273 (2004)
16. Rodríguez, J., Kuncheva, L., Alonso, C.: Rotation Forest: A New Classifier Ensemble Method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(10), 1619-1630 (2006)
17. Song Xu, Q., Zeng Liang, Y., Ping Du, Y.: Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. Chemometrics 18(2), 112-120 (2004)
18. Islam, M. M., Yao, X., Murase, K.: A constructive algorithm for training cooperative neural network ensembles. Neural Networks, IEEE Transactions on 14(4), 820-834 (july 2003)
19. Ardalani-Farsa, M., Zolfaghari, S.: Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks. Neurocomputing 73(13-15), 2540-2553 (2010)
20. Crone, S., Hibon, M., Nikolopoulos, K.: Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. International Journal of Forecasting 27(3), 635-660 (2011) Special Section 1: Forecasting with Artificial Neural Networks and Computational Intelligence Special Section 2: Tourism Forecasting.

Biomedical Signal Processing Using Wavelet-Based Neural Networks

Ever Juárez-Guerra[1], Pilar Gómez-Gil[2], Vicente Alarcon-Aquino[1]

[1] Department of Computing, Electronics, and Mechatronics,
Universidad de las Américas Puebla CP 72820, MEXICO
{ever.juarezga,vicente.alarcon}@udlap.mx
[2] Department of Computer Science,
Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro # 1, Tonantzintla, Puebla, México C.P. 72840
pgomez@acm.org

Abstract. Electroencephalography measures the electrical activity of the brain and represents a summation of post-synaptic potentials from a number of neurons. Electroencephalograms (EEG) are widely used in medicine for diagnostic and analysis of several conditions. In this work, we propose the construction of a system based on recurrent neural networks and wavelet analysis, able to analyze, detect and classify abnormalities in the brain such as Epileptic seizure using EEG as inputs. This work aims to develop novel algorithms to enhance the classification of the EEG signals and to improve the medical diagnosis. Self Recurrent Wavelet Neural Network (SRWNN) may be considered to classify the EEG signal and to improve the percentages of recognition in the classification between normal EEG and seizure EEG.

Key words: Electroencephalogram (EEG), Epileptic seizure detection, DWT, MODWT, Self Recurrent Wavelet Neural Networks (SRWNN).

1 Introduction

The transient and unexpected electrical disturbances of the brain result in a acute disease called Epileptic seizures. These seizures are seen as a sudden abnormal function of the body, often with loss of consciousness, an increase in muscular activity or an abnormal sensation [1]. Epilepsy is a neurological disorder affecting around 1% of the world population, where 25% of such patients cannot be treated properly by any available therapy [2]. The Electroencephalogram (EEG) signal has been a valuable clinical tool to assess human brain activities. In the last couple of years, the EEG analysis has been mostly focused on epilepsy seizure detection diagnosis [1], [3], [4]. The seizure detection problem is basically a classification between normal and seizure EEG signals. In recent years several models of artificial neural networks have been proposed, among these the Wavelet Neural Networks (WNN) that implement the wavelet processing as part of its operation through of the change of traditional transfer

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 23-32
Paper Received 20-02-2013 and Accepted 22-04-2013

functions as the sigmoid by wavelet functions. The combination of both theories seeks to exploit the features of analysis and decomposition of wavelet processing along with the properties of learning, adaptation and generalization of neural networks. In this work, we propose the design of a classifier, based on the use of Wavelet Transforms (WT) and WNN to detect and classify abnormalities in the human brain such as epilepsy. SRWNN may be also considered to detect these abnormalities [5].

1.1 Motivation

The human brain is obviously a complex system and exhibits rich spatio-temporal dynamics. Among the noninvasive techniques for probing human brain dynamics, electroencephalography provides a direct measure of cortical activity with millisecond temporal resolution. EEG signals involve a great deal of information about the function of the brain. Traditional methods rely on experts to visually inspect the entire length EEG recordings of up to one week, which is tedious and time-consuming [6]. The identification of right information extracted from an EEG of epilepsy patients that should be for the classification of seizures has recently attracted much attention. A classifier based on Recurrent Neural Networks can be an option to improve the classification of EEG signals due to its speed of convergence and less computational calculus.

2 Previous Work

In the last years, the EEG analysis was mostly focused on epilepsy seizure detection diagnosis. Most of the reported models are based on integration of computing technologies and problem solving paradigms, for example, neural networks [7], wavelets [8], logistical regression [6], histogram analysis [8], and chaos theory [9]. There are several approaches that have been proposed for the classification or detection of disorders of brain based on EEG signals. Next, we briefly describe some recent or relevant work where neural networks (NN) and wavelet analysis are the involved paradigms.

Tzallas et al. [4] demonstrate the suitability of the time- frequency *(t-f)* analysis to classify EEG segments for epileptic seizures and they compare several methods for *t-f* analysis of EEGs. Short-time Fourier transform and several *t-f* distributions are used to calculate the power spectrum density (PSD) of each segment. This analysis is performed in three stages: 1) *t-f* analysis and calculation of the PSD of each EEG segment; 2) feature extraction, measuring the signal segment fractional energy on specific *t-f* windows; and 3) classification of the EEG segment (existence of epileptic seizure or not), using Feed Forward Artificial Neural Network (FF-ANN). The method is evaluated using a benchmark EEG dataset of the University of Bonn [10] obtaining 89% of classification accuracy.

Gosh et al. [9] present a novel wavelet-chaos-neural network methodology for classification of EEGs of healthy (normal), ictal (seizure), and interictal patients. Wavelet analysis is used to decompose the EEG into delta (δ), theta

(θ), alpha (α), beta (β), and gamma (γ) sub-bands. This sub-bands are components of an EEG signal. Three parameters are employed for each segment of the EEG representation: standard deviation, correlation dimension, and largest Lyapunov exponent. The classification accuracies of the following techniques are compared: 1) unsupervised k-means clustering; 2) linear and quadratic discriminant analysis; 3) radial basis function neural network; 4) Levenberg-Marquardt backpropagation neural network (LMBPNN). A particular mixed-band feature space consisting of nine parameters and LMBPNN result in the highest classification accuracy of 96.7%. The EEG data used in this work are from of University of Bonn [10].

Anusha K. et al. [7] propose a NN based automated epileptic EEG detection system that uses FF-ANN incorporating a sliding window technique for pattern recognition. This work uses the database of University of Bonn, Germany [10]. The algorithm was trained with 50 segments of EEG, 25 cases of healthy patients and 25 of epileptic patients. The classification accuracy was 93.37% for distinguishing signals of normal patients and 95.5% for epileptic patients.

Shaik and Srinivasa [11] propose a classification system for epilepsy based on FF-ANN. A wavelet-based feature extraction technique is used to extract of features Energy, Covariance Inter-quartile range (IQR) and Median Absolute Deviation (MAD). Using the database of University of Bonn, Germany [10], a classification accuracy of 98% is reported.

Subasi et al. [6] present a method of analysis of EEG signals using WT and compare the classification obtained using Multilayer Perceptron Neural Network (MLPNN) and logistic regression (LR). They used lifting-based discrete wavelet transform (LBDWT) as a preprocessing method. A LR and a MLPNN classifiers were compared using EEG data owned by the authors. The result obtained of the classification accuracy of EEG signals by logistic regression was 89% and by MLPNN with Levenberg-Marquardt was of 92%.

3 Research Objective

This research project is focused on a study of a connectionist models to analyze, detect and classify abnormalities in the brain such an Epileptic seizure using EEG. Processing techniques such as Wavelet Transforms (WT) are used to analyze and detect Epileptic seizure. This work aims to develop a new algorithm based on recurrent wavelet networks to enhance the classification of the EEG signals and then improve the medical diagnosis. Our hypothesis is that a Self Recurrent Wavelet Neural Network (SRWNN)[12] may be considered to classify an EEG signal improving the classification accuracy compared with state of the art algorithms in the classification between normal and seizure EEG.

4 Research Methodology

We are taking the following steps to reach our research goal:

1. *Review the state of the art regarding classification of EEG.* This part consists of analyzing and understanding the work that have been proposed in the state of the art about classification of EEG. Also it is important to understand the characteristics and behavior of the EEG to select the databases of EEG that will be used to validate our proposal.
2. *Determine preprocessing strategies of EEG.* It is necessary to know the different methods to remove noise or artifacts from EEG signal and later to propose a suitable method to preprocess the EEG signals.
3. *Study the different techniques for processing of EEG.* This step consists on analyzing different techniques for processing and feature extraction of EEG and therefore propose the technique that will be used in this work.
4. *Modify the algorithm based on SRWNN to classify EEG signals.* Study the SRWNN and modify this algorithm to improve the classification accuracy of epilepsy on EEG signals. Notice that this step is part of our future work, therefore it is not reported here.
5. *Test the performance of the proposed algorithm.* Design representative experiments to test the performance of the new algorithm based on SRWNN in order to compare the results obtained with previous reported works and determine its advantages and disadvantages.

Fig. 1 shows the general block diagram of the proposed approach.

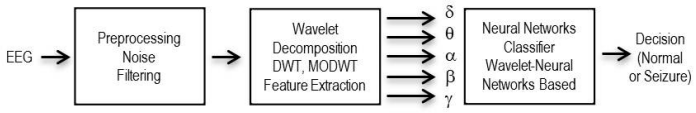


Fig. 1. General block diagram of research project

4.1 Originality and Main Contribution

In this research, the main contribution is a novel algorithm to enhance the classification accuracy of epilepsy on EEG signals by the implementation of a system using Wavelet-Based Neural Networks. The main characteristic of Wavelet-Based Neural Networks is that the activation function is changed by derivative functions of mother wavelet. Therefore, it is important to propose a suitable mother wavelet to enhance the classification of EEG signals. The selection of the wavelet must be related to the common features of the events found in real signals. In other words, the wavelet should be well adapted to the events to be analyzed [14]. This work also will consist on investigating others learning algorithms for SRWNN such as Metropoli Monte Carlo, genetic algorithms or optimization algorithms to obtain better results than previous works. Another contribution of this research consists in a suitable selection of the features extraction of each subbands of EEG signals providing the best information to enhance the classification

between normal and seizure EEG. EEG signals, like most biological signals, are inherently difficult to quantify and they may be characterized as non stationary signals. Therefore, it is necessary to select the appropriate characteristics that represent these non-stationary signals.

5 Preliminary Results

At this point, we have analyzed the state of the art regarding classification of EEG and we have studied the different techniques for preprocessing and feature extraction of EEG (steps 1, 2 and 3 of the research methodology). We are currently working on an analysis of classical processing of an EEG signal. Next we present the results obtained from a simple experiment related to the use of NN for classification of seizures. The objective of this experiment was to understand the whole process of EEG classification using a FF-ANN as classifier, instead of SRWNN, that it will be designed and analyzed later. This experiment is divided into three modules: preprocessing, feature extraction and classifier.

Experimental Data EEG. A database provided by the University of Bonn [10] is used in this experiment. These EEG data contain three different cases: 1)healthy, 2)epileptic subjects during seizure-free interval (interictal), 3)epileptic subjects during seizure interval (ictal) [10]. This database contains five datasets named: O, Z, F, N, and S. Sets O and Z are obtained from healthy subjects with eyes open and closed respectively. Sets F and N are obtained from interictal subjects in different zones of the brain; set S is gotten from an ictal subject [4]. Only the sets Z and S were used for the analysis reported here. This experiment was executed using Matlab 2010a and the Neural Network Toolbox Version 6.0.3.

Preprocessing. The aim of this block is to remove noise from EEG signal added to EEG signal during its recording. Since the sampling frequency of EEG records is 173.61 Hz, according to Nyquist sampling theorem [13], the maximum frequency of EEG should be in the range 0-86.81 Hz. Based on physiological research, frequencies above 60 Hz in EEG signal are considered as noise and can be neglected [8]. A Digital Butterworth low-pass filter of order 10 and cut off frequency of 64 Hz was used to eliminate these undesired frequencies. This filter was designed to meet with the following characteristics: 3 dB of ripple in the pass-band from 0 to 64 Hz, and at least 40 dB of attenuation in the stop-band [13].

Feature extraction. Two types of Wavelet Transforms were used for decomposition of the EEG signal: the DWT and the Maximal Overlap Discrete Wavelet Transform (MODWT)[14] using a second order Daubechies (Db2) and a fourth order Daubechies (Db4). Other wavelets may also be considered. WT is capable of "zooming-in" on short-lived high frequency phenomena and "zooming-out" on long-lived low frequency phenomena [14]. DWT is a WT for which the wavelets are discretely sampled. The DWT has some limitations it requires the sample size N to be an integer multiple of 2^J and the number N_j of scaling and wavelet coefficients at each level of resolution j decreases by a factor of two, due to the decimation process that needs to be applied at the output of

the corresponding filters. This limitations may introduce ambiguities in the time domain. The down-sampling process can be avoided by using the MODWT. The MODWT may be computed for an arbitrary length time series. Note, however, that the MODWT requires $O(Nlog_2N)$ multiplications, whereas the DWT can be computed in $O(N)$ multiplications. There is, thus, an increase in computational complexity when using the MODWT. However, its computational burden is the same as the widely used fast Fourier transform algorithm and hence quite acceptable [14]. A key advantage of MODWT over Fourier transforms is its temporal resolution: it captures both frequency and location information (location in time). Fig. 2 illustrates the decomposition of EEG sequence with four level MODWT extracting five physiological sub-bands (shown in yellow boxes).

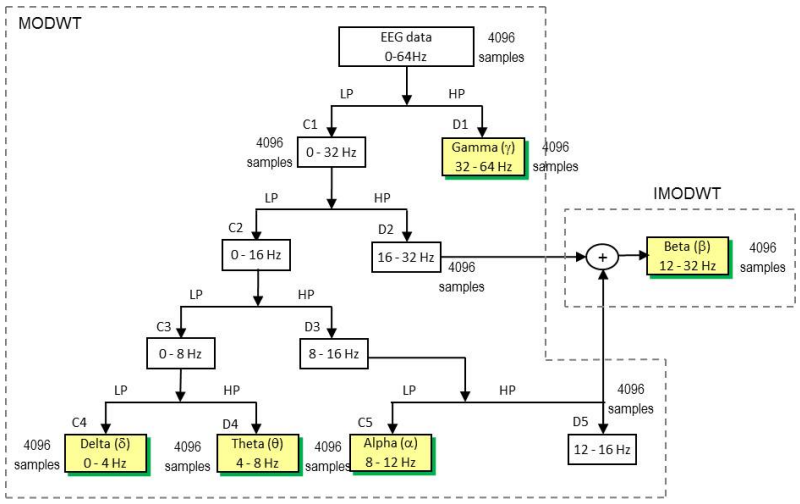


Fig. 2. Decomposition of EEG in physiological sub-bands by MODWT (yellow boxes). It shows the name of sub-bands and its respective frequency ranges.

Classification. According to Ravish [15], and Sunhaya [1] the delta and alpha sub-bands provided useful information to localize the seizure. Therefore in this experiment only these sub-bands of the EEG signal were used. Three statistical features (mean, absolute median and variance) of these sub-bands have been computed and input to a classifier based on a FF-ANN with 6 inputs, one hidden layer and one output. The classifier reported in [17] was used in this experiment. For training, 15 EEG segments from each set Z and S are used. For testing, 5 EEG segments from the same sets are used. Experiments were executed using 6 and 12 hidden nodes. The stopping criterion was specified to 0.01 Mean Square Error (MSE) and the learning rate was fixed at 0.5. The number of training epochs was fixed at 1000 and the activation function was a sigmoid. These values were experimentally chosen [17]. Fig. 3 a) and b) show a filtered EEG signal from a typical healthy and epileptic subject with its spectrum of

frequency (where it can be noticed the differences among the range of frequency of each EEG signal), respectively. Upper plots of Fig. 3 a) and b) represent the 4096 samples from an EEG segment of a healthy and epileptic subject (ictal), respectively. Notice that in both plots frequency components above to 64 Hz have been eliminated due to the Butterworth low-pass filtering. Fig. 4 and Fig. 5 show the decomposition by MODWT (Db2) of a segment of an EEG signal and its frequency spectrum of each sub-band from a typical healthy and epileptic subject, respectively. The graphs on the left side of Fig. 4 and Fig. 5 correspond to the Delta (0-4 Hz), Theta (4-8 Hz), Alpha (8-12 Hz), Beta (12-32 Hz) and Gamma (32-64 Hz) frequency sub-bands obtained by the MODWT (Db2) decomposition of a healthy and epileptic subject, respectively. The graphs on right side of Fig. 4 and Fig. 5 show the frequency components of each sub-band of a healthy and epileptic subject, respectively. A similar process is done using MODWT (Db4) and DWT (Db2 and Db4).

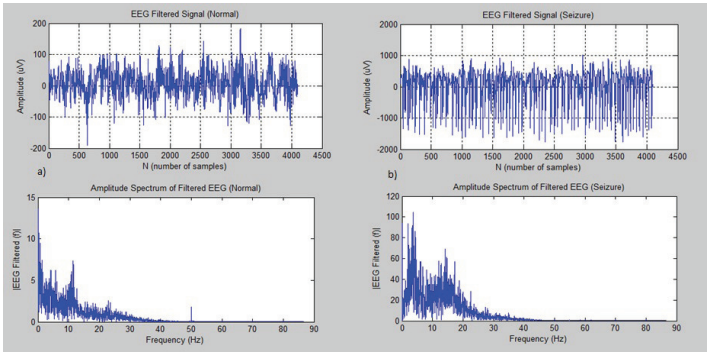


Fig. 3. Filtered signals EEG and its frequency spectrum of: a) Healthy subject, b) Ictal subject. Upper plots are samples from EEG signals and the lower plots show the frequency components of these EEG signals.

The results showed here are evaluated in terms of classification accuracy, sensitivity and specificity. Sensitivity (also called *the recall rate* in some fields) measures the proportion of actual positives which are correctly identified as such. Specificity measures the proportion of negatives which are correctly identified as such [12]. Sensitivity and specificity are calculated as:

$$sensitivity = \frac{TP}{TP + FN}(100\%) \tag{1}$$

$$specificity = \frac{TN}{TN + FP}(100\%) \tag{2}$$

where, TP (True positive) = correctly identified; FP (False positive) = incorrectly identified; TN (True negative) = correctly rejected and FN (False negative) = incorrectly rejected [12]. The results are compared with three published

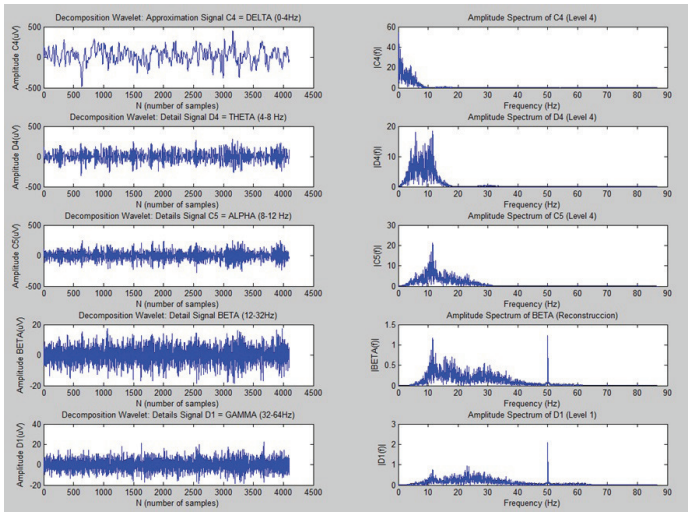


Fig. 4. Decomposition EEG signals by MODWT (Db2) of a healthy subject (normal). The graphs on the left side show the sub-bands obtained and the graphs on the right side show its corresponding frequency spectrum.

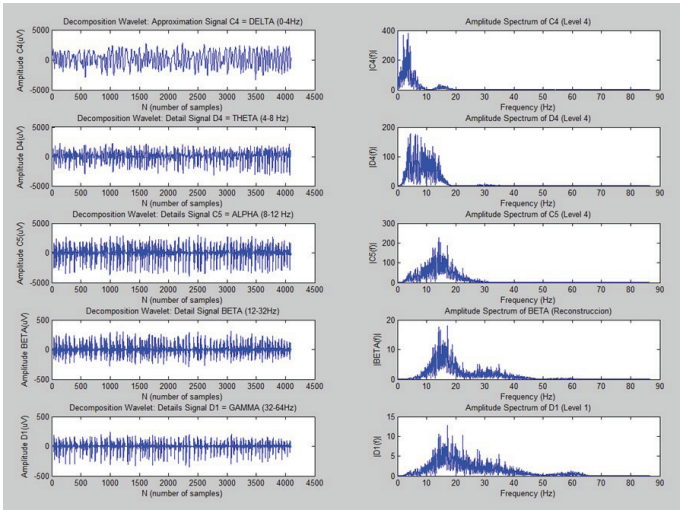


Fig. 5. Decomposition EEG signals by MODWT (Db2) of an ictal subject (seizure). The graphs on the left side show the sub-bands obtained and the graphs on the right side show its corresponding frequency spectrum.

works using the same databases, these are shown in Table 1. The best result obtained in this work was 90% of accuracy, using features calculated by MODWT (Db2) with 12 nodes in the hidden layer of the FF-ANN. The result reported in [11] was 98% of accuracy, which could be due to the fact that Shaik and collaborators used other characteristics better suited for the problem, or that they used more samples for training. They divided each segment of the database into 23 sub-segments (1 second). In this work we used the whole segment to calculate the characteristics.

Table 1. Result of the classifier of EEG signals

Authors	Parameters	Accuracy	Sensitivity	Specificity
Shaik et al. [11]	DWT Db4	**98.3**%	97.6%	98.5%
Gosh et al. [9]	DWT Db4	96.7%	—	—
Subasi et al. [6]	LBDWT Db4	92%	91.6%	91.4%
Proposed approach (6 hidden nodes)	DWT Db2	70%	100%	62.5%
	MODWT Db2	70%	75%	66.6%
	DWT Db4	70%	100%	62.5%
	MODWT Db4	80%	100%	71.4%
Proposed approach (12 hidden nodes)	DWT Db2	80%	100%	71.4%
	MODWT Db2	**90%**	**100%**	**83.3%**
	DWT Db4	80%	100%	71.4%
	MODWT Db4	90%	96.6%	83.3%

6 Conclusions

Diagnosing epilepsy is a difficult task requiring observation of the patient, an EEG, and gathering of additional clinical information. An artificial neural network that classifies subjects suffering an epileptic seizure provides a valuable diagnostic decision support tool for neurologists treating potential epilepsy. In this work, we propose the design of a new classifier based on wavelets and neural networks for identification of seizures events of epilepsy. It is expected that SRWNN classifier may obtain better results than previous reported works. The proposed classifier will use wavelet analysis as a tool for feature extraction. Here we presented the results by using two types of wavelets transforms to obtain features of EEG, and classified them using a FF-ANN. The features are extracted using the DWT and the MODWT in a whole segment of an EEG. Three features were extracted from delta and alpha sub-bands: mean, absolute median and variance. These six features are used to train a FF-ANN, which was able to discriminate among healthy and seizures EEG's in 90% of the samples in the testing set. Currently we are studying the specific characteristics of the SRWNN in order to adjust its design to this problem. We expect that SRWNN will obtain better results than FF-ANN. Future work also will be focused on investigating other possible training algorithms and a better feature selection.

References

1. Sunhaya, S. Manimegalai, P.: Detection of Epilepsy Disorder in EEG Signal. International Journal of Emerging and Development, Issue 2, Vol.2 (2012)
2. Engel J., Pedley T.: Epilepsy: A Comprehensive Textbook. Lippincott Williams & Wilkins, Philadelphia (1997)
3. Yuedong S.: A Review of Developments of EEG-Based Automatic Medical Support Systems for Epilepsy Diagnosis and Seizure Detection. J. Biomedical Science and Engineering. 4, 788–796 (2011)
4. Tzallas A.T., Tsipouras M.T., Fotiadis D.I: Epileptic Seizure Detection in EEGs Using Time-Frequency Analysis. IEEE Transactions on Information Technology in Biomedicine, Vol. 13 No. 5 (2009)
5. Sung J. Y., Jin B. P., Yoon H. C.: Direct Adaptive Control Using Self Recurrent Wavelet Neural Network Via Adaptive Learning Rates for Stable Path Tracking of Mobile Robots. In: Proceedings of the 2005 American Control Conference, 288-293.Portland (2005)
6. Subasi A., Ercelebi E.: Classification of EEG Signals Using Neural Network and Logistic Regression. J. Computer Methods and Programs in Biomedicine. 78, 87-99 (2005)
7. Anusha K.S., Mathew T.M., Subha D.P.: Classification of Normal and Epileptic EEG Signal Using Time & Frequency Domain Features Through Artificial Neural Network. In: International Conference on Advances in Computing and Communications. IEEE (2012)
8. Mirzaei A., Ayatollahi A., Vavadi H.: Statistical Analysis of Epileptic Activities Based on Histogram and Wavelet-Spectral Entropy. J. Biomedical Science and Engineering. 4, 207–213 (2011)
9. Ghosh D. S., Adeli H., Dadmehr N.: Mixed-Band Wavelet-Chaos-Neural Network Methodology for Epilepsy and Epileptic Seizure Detection. IEEE Transactions on Biomedical Engineering, Vol. 54, No. 9 (2007)
10. Andrzejak, R. G., Lehnertz, K., Rieke, C, Mormann, F., David, P., Elger, C. E. (2001): Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Phy. Rev. E, 64(6), 061907
11. Shaik J.H., Srinivasa K.: Epileptic Seizures Classification from EEG Signals Using Neural Networks. In: International Conference on Information and Network Technology. Vol 37 (2012)
12. García-González Y.: Modelos y Algoritmos para Redes Neuronales Recurrentes Basadas en Wavelets Aplicados a la Detección de Intrusos. Master thesis, Universidad de las Américas Puebla (2011)
13. Proakis J., Manolakis D.: Digital Signal Processing. Prentice Hall, 4th Ed. (2007)
14. Alarcón-Aquino V., Barria J.A.: Change Detection in Time Series Using the Maximal Overlap Discrete Wavelet Transforms. Latin American Applied Research: An International Journal, Vol.39, No.2 (2009)
15. Ravish D. K., Devi S. S.: Automated Seizure Detection and Spectral Analysis of EEG Seizure Time Series. European Journal of Scientific Research, Vol.68, Issue 1, (2012)
16. Xanthopoulos P., Rebennack S., Liu C., Zhang J, Holmes G. L., Uthman B. M., Pardalos P. M.: A Novel Wavelet Based Algorithm for Spike and Wave Detection in Absence Epilepsy. IEEE Int. Conf. on Bioinformatics and Bioengineering (2010)
17. Gómez-Gil P.: An Introduction to the Use of Artificial Neural Networks. Tutorial, available at: `http://ccc.inaoep.mx/~pgomez/tutorials/Gomez2009T-ANN.zip`

Recommendation of Process Discovery Algorithms: a Classification Problem

Damián Pérez-Alfonso, Raykenler Yzquierdo-Herrera, and Manuel Lazo-Cortés

University of Informatics Sciences, Havana, Cuba
{dalfonso,ryzquierdo,manuelslc}@uci.cu

Abstract. Process mining techniques extract knowledge from event logs of information systems. Process discovery is a process mining category, focused on discovering process models. The applicability and effectiveness of process discovery algorithms depend on event log's features. Selecting the right algorithms is a tough task due to the variety of variables involved and the complexity of obtaining logs features. To choose a suitable discovery algorithm the traditional approaches use empirical assessment. The metrics to perform this assessment are not applicable to all algorithms. Besides, empirical evaluation is time consuming and computationally expensive. The present paper proposes a new approach that, based on event log characteristics, recommends the discovery algorithms to be used. A new technique of sub-processes diagnosis is proposed for characteristics extraction. The recommendation procedure is formalized as a typical classification problem. This approach could be useful for large event logs and unclear processes analysis.

Key words: process discovery, process mining, classfication

1 Introduction

Information systems record in event logs the execution of supported business processes. Process mining involves discovery, conformance and enhancement of process starting from event logs. Performance evaluations, anomaly identification, compliance checking, among other kinds of analysis, need models that accurately reflect the actual execution of processes. This need has driven to the development of a variety of algorithms for discovering process models.

A process discovery algorithm is a function that maps an event log onto a process model, such that the model is "representative" for the behavior seen in the event log [1]. Noise, duplicate tasks, hidden tasks, non-free choice constructs and loops are typical problems for discovery algorithms [2]. Other problems are related to the mining of unstructured processes, commonly present in real environments [3]. Full or comprehensive solutions to the aforementioned challenges have not been submitted in literature. Thus, algorithms effectiveness depends on event log characteristics and their associated process.

The varying performance of discovery algorithms creates uncertainty during its application. Obtaining a quality model could require the use of several algorithms interchangeably, thus becoming a time consuming task. Selecting the

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 33-42
Paper Received 20-03-2013 and Accepted 22-04-2013

right algorithms is a hard task due to the variety of variables involved and the complexity of obtaining event logs properties.

A set of techniques for evaluating discovery algorithms has already been developed. Two kinds of metrics are used: metrics based on comparing the behavior in the discovered model with the behavior in the log, and metrics based on comparing the discovered model against a reference model related to the process [4]. Several techniques executes different discovery algorithms for an event log and evaluates resulting models using both kinds of metrics. Nevertheless, using this empirical evaluation approach for every event log is computationally expensive and time consuming. An approach that overcomes these shortcomings requires reference models [5]. However, reference models are not commonly available in contexts where process discovery is required. If there are reference models, it is unwise to assume that they reflect the actual execution of processes.

Studies that attempt to establish the algorithms with better performance under certain conditions have been published using the aforementioned empirical evaluation techniques [2]. But recognizing these conditions in real environments is a complex task. Also, the impact of each condition on model quality is not clearly defined yet. The existing metrics are not applicable to all algorithms due to modeling notation issues. Therefore, the actual use of these studies remains limited.

The aim of this paper is to establish the necessity and feasibility of a new approach to select discovery algorithms. In this paper, process discovery challenges are analyzed. Also, a critical review of main evaluation and selection techniques, so far proposed, is carried out. Through a novel technique of sub-processes diagnosis, it is possible to extract event log features such as: control-flow patterns, invisible tasks and infrequent behavior (noise). Therefore, it is feasible to construct a recommendation system of discovery algorithms starting from event logs characteristics. The recommendation procedure for this system is formalized as a typical classification problem.

The paper is structured as follows. Difficulties of process discovery are presented in the next section. Section 3 provides a literature review of techniques and approaches for evaluation and recommendation of discovery algorithms. In Section 4, the proposal of this paper is explained: firstly the factors to consider for a comprehensive mechanism of recommendation are established, usefulness of a new diagnosis technique for extracting log features is described, and finally, an outlook on some aspects for dealing with the recommendation of algorithms as a classification problem is projected. The last section is devoted to conclusions and outlines for future work.

2 Process Discovery Challenges

In order to properly select a discovery algorithm it is important to master the difficulties of process discovery. For a comprehensive and accurate recommendation all these difficulties and their influence on algorithm performance must be taken into account.

Heterogeneity of data sources from real environments, among other reasons, can lead to difficult cases for discovery algorithms [6]. Infrequent traces and data recorded incompletely and/or incorrectly can induce wrong interpretations of process behavior. Moreover, data provided by parallel branches and ad-hoc changed instances generates complex sequences on event logs, creating traces that are harder to mine.

Process structure is another source of challenges for discovery algorithms. Presence of control-flow patterns like non-free choices, loops (nested or not) and parallelism affect the discovery algorithms. For example, algorithms like α, α^+, $\alpha^\#$ and α^* do not support non-free choices [7]. On the other hand, DWS Mining and α^{++} can deal with non-free choice but cannot support loops [2].

The discovery of a process model requires that the event log contains enough information, i.e. has a level of completeness such that its traces are representative of process behavior. Completeness of event logs can be affected by absent information, i.e. existence of invisible tasks in the process. FSM Miner/Genet, β (Tsinghua α), α, α^+ and α^{++} are algorithms affected by invisible tasks.

Obtaining a quality model is another challenging aspect in process discovery. There are various metrics and approaches for estimating process model quality, though there is a consensus on the following quality criteria presented by Aalst [1]:

- *Fitness:* The model should allow the behavior present in the event log.
- *Precision:* The model should not allow a behavior that is completely unrelated to that present in the log.
- *Generalization:* The model should generalize the behavior present in the log.
- *Simplicity:* The model should be as simple as possible. Also referred as structure, is influenced by the vocabulary of modeling language.

These are competing criteria because there is an inverse relationship between generalization and precision. A too general model could lead to allow much more behavior than present in the log, also known as underfitting model. On the contrary, a too precise or overfitting model is undesirable.

The right balance between overfitting and underfitting is called *behavioral appropriateness*. The *structural appropriateness* of a model, on the other hand, refers to its ability to clearly reflect the performance recorded with minimal possible structure [8]. A quality model requires both behavioral appropriateness and structural appropriateness [9]. It can be appreciated that it is difficult to achieve a proper balance between the abovementioned quality criteria.

In real environments it is common to find unstructured processes. Many business processes are not orchestrated by workflow tools [3]. Furthermore, even when processes must be executed according to a designed process model, in practice, some room for flexibility is necessary for smooth operation of enterprise [10]. Both situations create unstructured processes, a challenging process type for discovery algorithms. The large number of alternative flows contained by logs from unstructured process complicates the detection of control-flow patterns and leads to complex and poorly understandable models.

3 Related work

In order to identify which discovery algorithm allows to obtain suitable models for particular situations, a set of techniques for algorithms evaluation have been developed. Performance of these algorithms is determined through evaluation of quality of obtained models. Defined quality metrics are grouped under two main methods [4]. One method compares the discovered model with respect to the event log and is called *model-log*. The other method, called *model-model*, assesses similarity between discovered model and a reference model of process.

Rozinat et al. devised an evaluation framework that allows end users to validate the results of the application of process mining, and researchers in this area to compare the performance of discovery algorithms [11]. The proposal combines the above mentioned evaluation methods.

The framework uses two approaches: assessment of the quality of discovered models through existing evaluation metrics and a *k-fold cross validation*, an evaluation technique from machine learning domain. The negative examples needed for the *k-fold cross validation* are obtained by generating a random event log. This approach for negative examples generation may include false negatives.

Use of this framework as a recommendation mechanism is not suitable owing to the cost involved on empirical assessments of discovery algorithms. The *k-fold cross validation* requires k executions for each algorithm to be assessed. In the other approach several executions are needed by each algorithm too, because of different inputs of evaluation metrics. The metrics used are not directly comparable to each other as they measure different aspects of quality at different levels of granularity and they are defined for different modeling notations [11]. This also limits the use of this framework for recommendation of discovery algorithms.

Following the *model-log* method, Ma proposes another evaluation framework [12]. The main novelty of this framework is the inclusion of a parameter optimization step using *k-fold cross validation*. The parameters optimization is a key issue since its values affect the obtained model and thus the performance of discovery algorithms [13]. Negative examples are used in this framework for the evaluation stage and parameters optimization.

For the Ma framework, the empirical evaluations cost is also the main shortcoming for its use as a recommendation tool. In the experiments with complex event logs, the negative examples generation through AGNES [14] created serious performance problems [12]. This proposal has also limitations related to the modeling formalism supported by the selected metrics, so it is not applicable to any discovery algorithm.

De Weerdt et al complement the *model-log* evaluation method with a comprehensibility assessment of the models obtained [2]. Comprehensibility is associated to the quality criteria called simplicity. In [2] the performance of seven discovery algorithms was analyzed on real and artificial logs. Using statistical techniques was concluded that complexity of event log has an important impact on the evaluated quality criteria. Also, were found remarkable differences between assessment on artificial logs and assessment on real logs.

Some relationships found among simple event log features and the results of the discovery algorithm were highlighted in these paper. Nevertheless, these relationships must be quantified in order to be used for algorithm recommendation. It is also required to include the impact of more complex features such as noise, lack of information and completeness.

All evaluation techniques analyzed so far, apply mainly to algorithms that can discover models on Petri nets or are transformable to Petri nets, due to the preponderance of metrics associated with this notation. This limitation excludes algorithms such as Fuzzy Miner [15] and techniques like pattern abstractions or sequence clustering, which are particularly useful for discovering unstructured processes, a context where algorithms that generate Petri nets are ineffective [2]. Additionally, in real environments where process models are not specified or do not reflect the actual processes execution, the *model-model* evaluation method is not applicable. Moreover, selecting a discovery algorithm for a given situation based on empirical evaluation, involves time and resource consumption for each of the algorithms chosen as a possible solution.

3.1 Beyond empirical evaluation

Wang et al.'s [5] work is a major effort to minimize the empirical evaluation problem in selection of discovery algorithms. The proposed framework is based on selecting reference models of high quality and building from these a regression model to estimate the similarity of other process models. This approach is comprised of a learning phase and a recommendation phase.

With the structural characteristics of the significant references models and the similarity values obtained after empirical evaluation of algorithms, a regression model is constructed during the learning phase. In the recommendation phase the reference models features are extracted in order to predict the similarity results using the regression model. Starting from the estimated similarity values the ideal algorithm for discovering the processes associated to the reference model is proposed.

The experiments conducted using this recommendation solution show encouraging results [5]. The evaluation over a set of 621 models (including artificial and real models) achieved more than 90 % accuracy in the recommendations. However, this approach involves some requirements that severely limit its application.

The main constraining requirement is the need of reference models for the evaluation and prediction. In multiple real-world environments, where discovery algorithms need to be applied, the process models are not described or are inconsistent and/or incomplete. Weng et al.'s approach assumes that the actual execution of the processes keeps a close relationship with their reference model. Therefore, in contexts where features of the actual logs differ from logs artificially generated by the reference models inexact results can be expected. The construction of regression model from model features discards issues like noise, lack of information and completeness of event logs, which have a significant impact on the performance of discovery algorithms.

Last but not least, the proposed framework has been conceived for using labeled Petri nets as the modeling language. Therefore, it is only applicable in environments where the models are specified using Petri nets or equivalent notations. This constraint also implies that the framework could only recommend algorithms based on Petri net or equivalent notations, excluding algorithms such as the Fuzzy Miner. As mentioned before, this algorithm is especially useful in contexts where discovery algorithms that generate Petri nets are inefficient.

An alternative direction to solve the algorithm selection problem is offered by Lakshmanan and Khalaf [16]. The authors construct a decision tree starting from the comparison of five discovery algorithms. The comparison establishes the algorithms potentialities to tackle challenges like: invisible and duplicate tasks, loops, parallelisms, non-free choice and noise.

Nevertheless, it is not specified how to identify the presence of challenging situations in the process to mine. While recognition of complicated control-flow patterns can be performed from a reference model, this implies drawbacks already discussed. Moreover, identifying invisible tasks, duplicate tasks, noise, loops, parallelisms or non-free choice from an event log is far from trivial. In general, this study provides some important theoretical elements but lacks direct practice applicability due to the required information type.

4 Proposal

Taking into account the identified challenges for discovery algorithms and the aforementioned limitations of existing solutions for selection and recommendation, it can be established that the conception of a comprehensive mechanism for recommendation of discovery algorithms should consider the following factors:

1. The event log is the main information source that is available in all environments for process characterization.
2. The peculiarities of event log such as noise, lack of information, completeness and log size (number of event classes, number of traces, etc.) must be considered in addition to process characteristics.
3. The process characteristics and the event log peculiarities are not always fully identifiable.
4. There should be no limitations regarding modeling notation of the discovery algorithms to recommend.
5. The process discovery goal may require to reinforce some of the desired quality criteria in the model to be obtained.

Considering factors 1 and 2 there is a need for techniques capable of extracting information from an event log that could be useful for the recommendation. A new technique has been developed to obtain relative values of event log completeness starting from certain basic assumptions [17]. This technique employs statistical estimators and is implemented as a ProM plug-in. A new diagnostic tool based on sub-processes decomposition is specially useful to identify other event log peculiarities and process characteristics.

4.1 Sub-process decomposition

A new technique for estimating the lack of information in the event logs has been developed [18]. This technique is able to recognize the control-flow patterns present in an event log starting from its trace alignment [19]. The identified control-flow patterns allow to decompose the process into sub-processes. The decomposition is represented by a tree named tree of building blocks.

The sub-processes are the leaves in the tree of building blocks. Each leaf shows the trace alignment of the sub-processes and the frequency of the different cases. This information facilitates the identification of cases that potentially represent noise in each of the sub-processes. Based on the tree of building blocks this tool identifies patterns of lack of information and estimates the associated invisible tasks. As can be seen above, this tool identifies, from an event log, a set of characteristics that affect the performance of the discovery algorithms.

It can also be particularly useful to segment large and/or complex event logs in a meaningful way. The sub-processes generally possess different features: presence of different control-flow patterns, different types of trace frequency, existence of patterns of lack information. These features determine the effectiveness of discovery algorithms, as it has already been explained. Therefore, it is suggested to perform the recommendation for each one of the sub-processes because even if they belong to the same event log, different sub-processes may require different discovery algorithms.

4.2 Recommendation as a classification problem

Classification is the problem concerning the construction of a procedure that will be applied to a continuing sequence of cases, in which each *new case* must be assigned to one of a set of *pre-defined classes* on the basis of *observed attributes or features* [20]. The recommendation of discovery algorithms can be expressed in terms of a classification problem. An event log on which is necessary to recommend a discovery algorithm is considered as a *new case* to be classified. The recommended algorithm is the *pre-defined class* to be assigned to an event log based on its *observed features*. This kind of classification procedure where the true classes are known has also been variously termed as pattern recognition, discrimination, or supervised learning [20].

Typifying the recommendation of discovery algorithms as a classification problem opens up the way to the application of techniques developed in a long-standing knowledge area. Techniques from the *classification* area have already been used for the recommendation of discovery algorithms [5]. Nevertheless, for the conception of a comprehensive recommendation mechanism that overcomes the limitations of existing solutions, it is important to incorporate to this classification problem the aforementioned factors. The recommendation mechanism proposed can be observed in Fig. 1.

According to the first of the aforementioned factors, the features for the classification of the *new case* should be extracted just from the event log. It should be noted that the characteristics mentioned in the second factor differ in

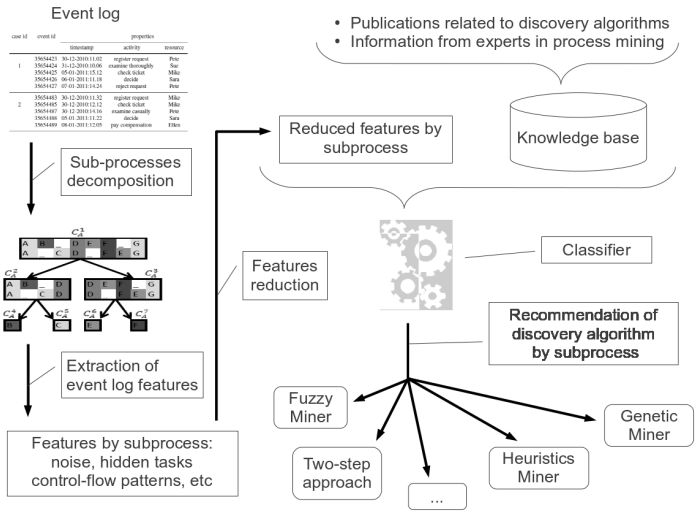


Fig. 1. Recommendation of discovery algorithms through sub-processes decomposition.

their values scales because in this classification problem the *cases features* have different orders of magnitude.

To solve a classification problem the design of the classifier is an essential issue. Roughly speaking, there are three different approaches for designing a classifier. The first approach is based on the concept of similarity, the second one is a probabilistic approach and the third approach is to construct decision boundaries directly by optimizing certain error criterion [21]. Selecting the right approach and designing an efficient classifier is a complex task that exceeds the scope of this paper. However, taking into account factor 3, it is useful to stress that the classifier which will be designed for the recommendation of algorithms should be able to deal with incomplete information about *cases features*.

A major challenge for this classification problem is the building of a knowledge base. Useful information for the knowledge base can be provided by a set of publications and experts in the field of process mining. Published results about assessment of discovery algorithms using traditional approaches are a good starting point. Nevertheless, results tied to existing quality metrics do not cover all the algorithms due to modeling notations issues. Considering the factor 4, information about performance of algorithms that cannot be assessed using quality metrics is needed. Therefore, the building of the knowledge base for algorithms recommendation opens up the opportunity for further research.

Several alternatives can be assessed to incorporate the factor 5 into the classification problem. To include in the classes definition the combination of the discovery algorithm and the quality criterion to reinforce is an alternative. To include the quality criterion to reinforce as part of the characterization of cases to classify could be another alternative.

Performing the recommendation of discovery algorithms through classification requires further research. The existing knowledge in the process mining area should be recovered and structured it on one of the existing representations for that purpose. An efficient classifier from existing approaches must be designed. The feature selection procedure to apply is needed. Nevertheless, in order to formalize the classification solution some insights were provided taking into account the aforementioned factors for the conception of a comprehensive recommendation mechanism.

5 Conclusions and Future Work

Characteristics such as noise, duplicate tasks, hidden tasks, non-free choice constructs and loops could affect the performance of discovery algorithms. Current approaches that select discovery algorithms based on empirical assessments are computationally expensive and time consuming. Other approaches presuppose the existence of reference models; however, there is a low probability of having available reference models. In general, the approaches based on assessment have shortcomings that are induced by the existing metrics. Besides, the shortcomings are related to the modeling notation of the discovery algorithms.

This paper presented a new approach that based on event log characteristics recommends the discovery algorithms to be used. Important factors to be considered in the conception of a comprehensive mechanism to recommend discovery algorithms were declared in the proposal. The technique for sub-processes diagnostic described in Section 4.1 would be especially useful for recommendation based on event logs characteristics.

The recommendation of discovery algorithms can be treated as a typical classification problem. Basic ideas to formalize that classification problem were presented. To solve the recommendation of discovery algorithms as a classification problem further research is required in order to build up a knowledge base, to select its most suitable structure, to apply an effective features selection and to design the classifier. A great starting point for those further research works are the available techniques in a long-standing knowledge area like classification.

The recommendation of discovery algorithms using subprocesses and based on event logs characteristics could be useful for process mining projects in enterprises with large and complex event logs.

References

1. van der Aalst, W.M.P.: Process Mining. Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg, Dordrecht, London et. al (2011)
2. De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A multi- dimensional quality assessment of state-of-the- art process discovery algorithms using real- life event logs. Information Systems **37** (March 2012) 654–676
3. Desai, N., Bhamidipaty, A., Sharma, B., Varshneya, V.K., Vasa, M., Nagar, S.: Process trace identification from unstructured execution logs. In: Services Computing (SCC), 2010 IEEE International Conference on. (2010) 17–24

4. De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A critical evaluation study of model-log metrics in process discovery. Volume 66 LNBIP of 8th International Workshops and Education Track on Business Process Management, BPM 2010., Hoboken, NJ (2011)
5. Wang, J., Wong, R.K., Ding, J., Guo, Q., Wen, L.: Efficient selection of process mining algorithms. IEEE Transactions on Services Computing **99**(1) (2012) 1–1
6. Ly, L.T., Indiono, C., Mangler, J., Rinderle-Ma, S.: Data transformation and semantic log purging for process mining. In: 24th International Conference on Advanced Information Systems Engineering (CAiSE'12). LNCS, Springer (2012)
7. Van Dongen, B., Alves de Medeiros, A., Wen, L.: Process mining: Overview and outlook of petri net discovery algorithms. In: Transactions on Petri Nets and Other Models of Concurrency II. Volume 5460 of Lecture Notes in Computer Science., Springer (2009) 225–242
8. van der Aalst, W.M.P., Rubin, V., Verbeek, H., Van Dongen, B., Kindler, E., Günther, C.: Process mining: A two-step approach to balance between underfitting and overfitting. Software and Systems Modeling **9**(1) (2010) 87–111
9. Rozinat, A., van der Aalst, W.M.P.: Conformance testing: Measuring the fit and appropriateness of event logs and process models. In: Third International Conference on Business Process Management(BPM 2005), France (2006) 163–176
10. Mieke Jans, Michael Alles, Miklos Vasarhelyi: Process mining of event logs in internal auditing: A case study. (2011)
11. Rozinat, A., Medeiros, A.K.A.d., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: Towards an evaluation framework for process mining algorithms. BPM Center Report (2007)
12. Ma, L.: How to Evaluate the Performance of Process Discovery Algorithms. Master thesis, Eindhoven University of Technology, Netherlands (2012)
13. Weijters, A.: An optimization framework for process discovery algorithms. In Stahlbock, R., ed.: Proceedings of the International Conference on Data Mining, Las Vegas Nevada, USA (2011)
14. Goedertier, S., Martens, D., Vanthienen, J., Baesens, B.: Robust process discovery with artificial negative events. Journal of Machine Learning Research **10** (2009)
15. Günther, C., van der Aalst, W.M.P.: Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In: Business Process Management. Volume 4714 LNCS of 5th International Conference on Business Process Management, BPM 2007., Brisbane (2007) 328–343
16. Lakshmanan, G., Khalaf, R.: Leveraging process mining techniques to analyze semi-structured processes. IT Professional **99**(PrePrints) (2012) 1–1
17. Yang, H., van Dongen, B.F., ter Hofstede, A.H.M., Wynn, M.T., Wang, J.: Estimating completeness of event logs. (2012)
18. Yzquierdo-Herrera, R., Silverio-Castro, R., Lazo-Cortés, M.: Sub-process discovery: Opportunities for process diagnostics. In Poels, G., ed.: Enterprise Information Systems of the Future. Number 139 in Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2013) 48–57
19. Bose, R.P.J.C., van der Aalst, W.M.P.: Process diagnostics using trace alignment: Opportunities, issues, and challenges. Information Systems **37**(2) (April 2012) 117–141
20. Michie, D., Spiegelhalter, D.J., Taylor, C.C., Campbell, J., eds.: Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River, NJ, USA (1994)
21. Zheng, L., He, X.: Classification techniques in pattern recognition. In: WSCG, conference proceedings, ISBN. (2007) 80–903100

Analysis of Perceptual Models Based on Visual Cortex for Object Segmentation in Video Sequences

Juan Alberto Ramirez-Quintana, Mario Ignacio Chacon-Murguia

Visual Perception Applications on Robotic Lab
Chihuahua Institute of Technology, Chihuahua, Chih, Mexico
jaramirez@itchihuahua.com, mchacon@ieee.org

Abstract. Visual perception capacities provide us with the ability to recognize objects from the interpretation of shape, color, orientation and motion features. The mechanisms in the visual cortex that allow the interactions between those visual features have been formalized in neurocomputational models and Artificial Neural Networks. In this paper, we propose a method based on perceptual models of visual cortex to analyze color, texture and motion in video sequences oriented to object segmentation. The results of the methods inspired in the behavior of the visual cortex have shown coherent object segmentation in videos with real scenes.

Keywords. Receptive fields, Visual cortex, video segmentation.

1 Introduction

Visual perception is a set of capacities of the brain that allows for interpretation of the information generated by the light reflected on object surfaces. These capacities lead to divide a scene in visual elements, with the aim to recognize coherent objects, using features like color, shape, orientation and motion [1]. There are many neurocomputational models and Artificial Neural Networks (ANN) based on the behavior of the visual cortex that attempt to explain the perception process. These models could be classified into three kinds: Adaptive Resonance Theory (ART), Pulse Neural Networks (PuNN) and Self-Organized Models (SOM). Among the ART-based models, there are some inspired in the behavior of different types of neurons in the visual cortex and their interaction between object recognition [2] and motion [3]. Pulse Networks can be classified according to their dynamic mechanisms of coupled oscillations [4] and integrate and fire networks like Spiking Neural Networks (SNNs) [5]. Coupled oscillations are models involving perceptual theories that describe the nervous system as a set of oscillations with a temporal dynamic that form groups of neurons. The Laterally Excitatory Globally Inhibitory Oscillator Network (LEGION) is a network used in image segmentation and scene analysis based in coupled oscillations [4][6]. The SNNs are dynamic models inspired in the membrane

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 43-52
Paper Received 19-02-2013 and Accepted 22-04-2013

potentials of the neurons and they have been used to model different perceptual mechanism like in [7][8]. SOM models are inspired from the hypothesis that the synaptic weights in visual cortex are organized through self-organization based on the input stimuli. The most popular model of the SOM networks is the Kohonen Network, which have been used in several applications for pattern recognition and simulation of perceptual mechanism given in the visual cortex [9]-[11].

Visual perception theories have been a useful inspiration to develop methods for pattern recognition applied in computer vision, because it provides insights that allows the recognition of visual patterns in complex scenarios like the illusions shown in figure 1. Many works based on different approaches like ANNs or probabilistic methods try to relate the visual perception theories with image processing and computer vision applications. However, some of these works tend to be very specific in the application of certain principles, or they use unreal scenes for testing, and the processing times are not plausible for real time video analysis. Therefore, in this work we propose an analysis of neurocomputational models based on visual cortex theories and ANN, with the aim to design a real-time method using visual perception theories supported by visual cortex models for static or moving object segmentation in video sequences in real scenarios.

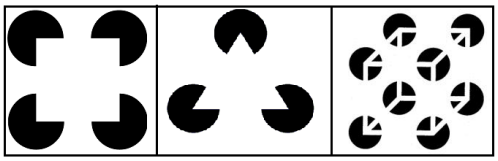


Fig. 1. Common illusions analyzed in visual perception theories.

We first present a state of the art of models based on the visual cortex that explain perceptual mechanism in object detection and motion. Then, we simulate the perceptual mechanism given in those models with ANN, and based on the simulations we describe a real-time method to perform coherent object segmentation in video sequences.

The rest of the paper is organized as follows: Section II presents the fundaments of the visual cortex. Section III analyses the models used in the segmentation process. Section IV describes the main aspects in the design of the segmentation methods and section V reports the final discussion.

2 Fundaments of Object Detection in the Visual Cortex

The visual cortex is the section of the brain that implements the main tasks of the visual perception process. This process starts in the retina, which acquire the color information with the cones, and the illumination changes with the rods. These light signals are converted in electrical signals, and they are distributed in the Retina Ganglia Cells (RGC) in a set of Receptive Fields (RF) calling ON and OFF. The RGCs have a set of axons that form the optic nerve and they are extended to the Lateral Geniculate Nucleus (LGN), which pass the information to the visual cortex

and filter the useless information. The RFs are sensible to changes light-dark, wavelength Red-Green (R/G), Green-Red (G/R) and Blue-Yellow (B/Y) [12][13], all distributed in the channels ON, OFF, as illustrated in figure 2. They are modeled with Difference of Gaussians functions, DoG, where ON has a positive center Gaussian with variance less than the variance of the negative Gaussian, OFF has a negative center Gaussian with a smaller variance than the variance of positive Gaussian.

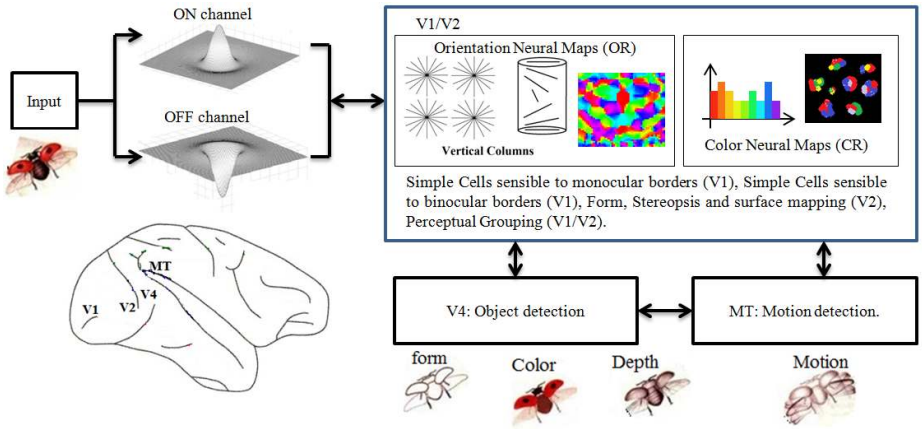


Fig 2. Visual cortex parts for object and motion detection.

In the primary visual cortex (V1) and the visual area 2 (V2), the information of the RF is mapped in a set of neurons that are sensible to certain visual features. A common example of these neurons are the vertical columns [14][15], which are cells that describes the orientation patterns of the visual information under a retinotopic structure, and they are calling Orientation Map (OR). V1 and V2 have other similar maps sensible to color features (CR map) and ocular dominance (DO map) [16]. Those maps avoid the redundancy and develop the perception process forming features like shape, color, illumination changes, orientation and depth. According to the literature, it is consistent that features like color, orientation and depth are the inputs to the visual area 4 (V4), which is associated to the object detection and the medial temporal cortex (MT) is associated to motion perception. As figure 2 shows, the visual cortex has feedback loops in all its parts, which is an important item in the formation and recognition of visual patterns. The visual cortex has other cortical parts not analyzed in this work.

3 Scheme of the methods obtained from the perceptual models.

Visual perception of dynamic objects by the visual cortex involves a feedback between object detection and motion perception (V4 and MT). Therefore, with the analysis of those visual cortex parts, we propose the designing of a method based on the interactions between V1, V2, V4 and MT to achieve object segmentation, where the objects are classified in static and dynamics. The objects that can change its positions in the scene are dynamic objects, while those objects that remain in the same

position in the scene during the video sequence are the static objects. The figure 3 shows a scheme of the proposed method, where the LGN processes the information in the RF channels with DoG. In the next step, V1/V2 processes different features used in the modules V4 and MT achieves the objects segmentation function. The videos used in the experiments were acquired with stationary cameras. The next subsections describe the visual cortex models used to design the segmentation methods.

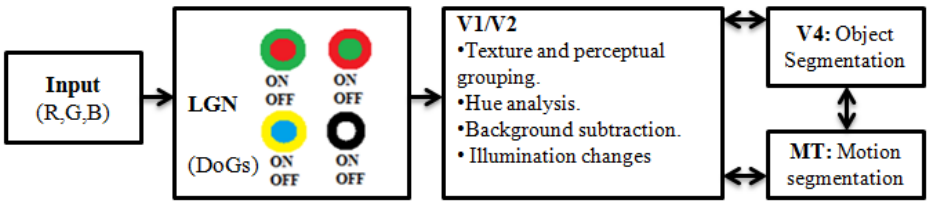


Fig 3. Scheme of the method to static and dynamic object segmentation.

3.1 V1 Feature extraction model

According to [12], V1 and V2 extract different features from visual patterns, using a set of neurons that organize the weights to form the cortical maps OR, CR, DO and illumination changes. To represent those cortical layers, we analyzed a set of models of the family of Laterally Interconnected Synergetically Self-Organizing Map (LISSOM) [15], which are developed from the theories on visual cortex behavior. The aim of LISSOM is to simulate the simple cells of the retina, LGN, V1 and in occasions V2 that activate the cortical maps. The basic LISSOM model (Fig. 4) is used to train the OR map, which describes the main orientations of a visual stimulus through time. The map is codified with hue levels like in the figure 4.

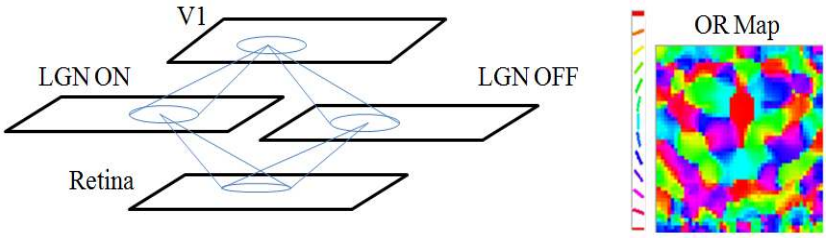
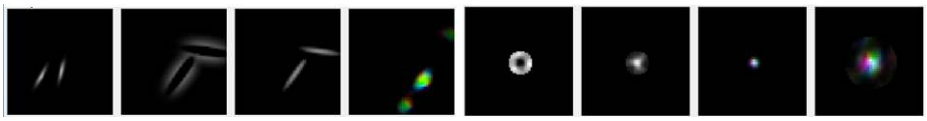


Fig 4. Architecture of the LISSOM model to activate the OR map, which is codified in the hue space according to the orientation bar shown.

Figure 5 shows a simulation of the LISSOM model, where the retina has two Gaussian patterns, the LGN ON/OFF have fixed weights given by DoG and the Figures 5b y 5c show the response of the LGN, which have afferent weights connected with V1. The cortical activation of V1 is based on the afferent weights, the lateral excitatory weights E and the lateral inhibitory weights L. The hebbian learning is used to update all the weights. The complete model is documented in [15]. The weights converge after approximately 10000 iterations (Figures 5e-5h). The weights model the OR map, V1 and both cortical maps are combined forming the response

shown in figure 5d where V1 activates the orientation patterns of the visual stimuli in the actual iteration.

From among the LISSOM models, the Tricromatic LISSOM model (TLISSOM) is a model that describes how V1 and V2 stimulate the organization of the neural color map (CR). Another LISSOM model is Perceptual Grouping LISSOM (PGLISSOM), which is a set of SNNs that construct the OR map through pulse mechanism. This model drives the weight like the basic LISSOM model and solves perceptual grouping problems like the *kanizsa* triangle [15].



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. LISSOM model response. (a). Input patterns, LGN response (b). OFF and (c). ON. (d). V1 final response. (e), (f). Afferent weights. (g). *E* weights. (h). *L* weights.

Among the LISSOMs models, the SOM Retinotopic Map (SOMRM) [15], represents the cortical mechanism of LISSOM, but SOMRM is similar to the *Kohonen* network. The figure 6a shows the architecture of the SOMRM, which have two layers; the retina and V1. The retina is a set of K receptors that sends the input pattern to V1, which is a set of neurons with synaptic weights in the interval of [0,1]. The neurons W_{ij} are represented by vector weights $W_{ij}=\{\omega_{1ij}, \omega_{2ij}, \omega_{kij} \ldots, \omega_{Kij}\}$ and according to [15], the initial response (η_{ij}) is given by the dot product between the neurons W_{ij} and the input I_k. The competition process to select the winner neuron in the SOMRM is given by the highest value in η_{ij}. Moreover, this competition process is different to the selection of the winner neuron in the Kohonen network, where the winner is given by the minimum value of the Euclidian distance between the input and the weights. According to [15], the learning of the SOMRM is based on the normalized hebbian rule determined as:

$$\omega_{k,ij}^{t_s+1} = \left(\omega_{k,ij}^{t_s} + \alpha I_k \beta_{ij}\right) \Big/ \sqrt{\sum_k \left(\omega_{k,ij}^{t_s} + \alpha I_k \beta_{ij}\right)^2} \tag{1}$$

where, t_s are the iterations of the SOMRM, α is the learning rate, which fall to zero if $t_s \rightarrow \infty$. β_{ij} is the neighborhood function computed as:

$$\beta_{ij} = \eta_{i_w j_w} \exp\left(-\frac{(i_w - i)^2 + (j_w - j)^2}{\sigma_\beta^2}\right) \tag{2}$$

where (i_w, j_w) is the index of the winner neuron, σ_β is the neighborhood radio which falls to 0.5 if $t_s \rightarrow \infty$ [15]. Then, the SOMRM has two learning parameters; α, that defines the learning of all neurons in the hebbian rule and β_{ij}, that defines the learning capability per neuron based on the distance of each neuron to the winner.

The input pattern in the retina is a Gaussian that changes its position in each iteration like in figure 6b. In initial conditions, the weights in the SOMRM has random values in the interval [0,1] like in figure 6c, also $\alpha > 0$ and $\sigma_\beta > 0$. If $t_s \rightarrow \infty$, the neurons tend to learn the input pattern I_k, i.e., the neurons learn the Gaussian pattern

on different positions as illustrated in figures 6d y 6e. Therefore, the SOMRM network is a simulation of V1 based on SOM, where each neuron develops capability of spatial selectivity, forming topographic maps in order to represent a pattern that changes its position in the retina surface.

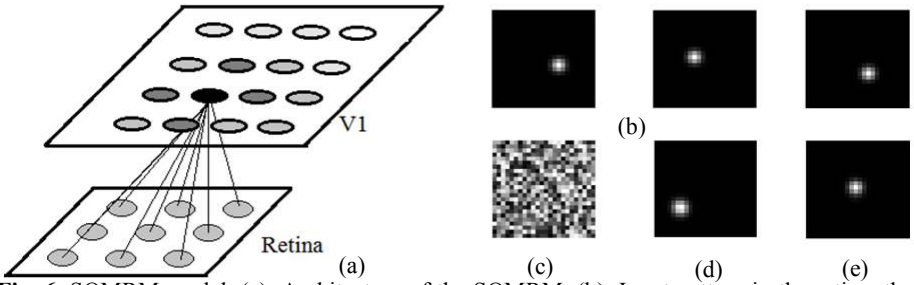


Fig. 6. SOMRM model. (a). Architecture of the SOMRM. (b). Input pattern in the retina, that changes its position. The size of the retina is *K=576* (*24x24*). (c). Neuron initial condition weights. (d) and (e). Neurons in the final iterations, where (d) is a neuron positioned in the border and (e) is a neuron positioned in the center.

3.2 Feature extraction for object detection

As we mentioned before, neurons in V1 and V2 are sensible to features of color, orientation and illumination changes. According to [13], the self-organizing interactions in V1 develop the modeling of the maps OR, CR.

The color analysis for the proposed scheme is based on TLISSOM model documented in [13]. This model describes how the training of the neural map CR is performed by the self-organization of V1, V2, and the processing of color RFs G/R, R/G and B/Y. Therefore, the method for color analysis in the proposed method is based on color RFs and cortical layers with self-organized process. The input of the method is a frame of a video sequence in the RGB color space. The RFs are represented with the sum of two convolutions between a Gaussian with a color channel. For example RF G/R ON is represented from the sum of convolutions between a positive center Gaussian with the red channel (R) and a negative surround Gaussian with higher variance with the green channel (G). In the channel G/R OFF, the sign of the center Gaussian is negative, and the sign of the surround Gaussian is positive. Despite the input of RFs is a RGB frame, the RF response is sensible to Hue, Saturation and Value (HSV) information of the frame. For example, as figure 7, the RFs are variant to H and S, but invariant to V changes (the response of the RF to the images in figures 7a y 7b are the same). The RF response is combined in V1, which is a self-organized cortical layer represented with an ANN based on SOMRM to produce four color channels (Red-Yellow-Green-Blue), all invariant to V changes. Those color channels are analyzed with their histograms to simulate the neural color map CR, which represent the activations in V2 that form the color groups. This method is inspired on TLISSOM that forms the color groups combining CR with the SOMRM and the color RFs is calling TRSOM and it achieves the object segmentation function using texture information.

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 7. RF Activation of the images with low bright (a), high bright (b). (c) and (d) RF Channels G/R. (e) and (f) Channels R/G.(g) and (h) Channels B/Y.

The texture and shape information is commonly analyzed with the orientation feature. Therefore, the proposed scheme uses the mechanisms focused to develop the orientation feature in the visual cortex for texture-shape analysis. For this feature, we used pulsed neural networks (PuNN), because they are based on the visual cortex behavior and they are commonly applied in image segmentation. From among the PuNN, the models LEGION [4][6] and PCNN [17] have been successfully applied in scene and texture segmentation. Therefore, we used the PCNN and the pulses were codified in time signatures defined in [17] which can perform the same neuronal synchronization as the coupled oscillations. Figure 8c illustrates an example of image segmentation with the PCNN defined in [17], where the time signature is calculated as:

$$SY_{ij} = \sum_{n=1}^{N} Y_{ij}(n) \tag{3}$$

where SY_{ij} is a time accumulation of the output pulses $Y_{ij}(n)$ in each iteration, N is the final iteration. Then, based on the PCNN time signatures and the PGLISSOM pulse mechanism [15], we design a pulsed model based on PCNN, where the input is a grayscale image. The RFs ON and OFF are given by DoG and the response of LGN is the convolution between the input and each RF channels. Then, LGN response pass to V1, which is represented by a set of Orientation Filters that represent the vertical columns in OR. To simulate the membrane actions potential in V1 we use the time signatures of the equation 3, where the pulses were modeled with a PCNN[17]. Figure 8d shows the proposed model, denominated RF-PCNN. In order to test the plausibility of the model with visual perception theories, we used two images commonly analyzed in the perception literature. Figure 8e shows an image with the *Hermann* illusion, where there is an optical illusion characterized by false blobs in the intersections of the grids. The strong suppression of the ON channel in the intersections causes this illusion [18]. Figure 8f shows an illusion where the observer could find a word if he/she tries to shrink the eyelids. When the RF-PCNN processes both images, the model obtains output patterns consistent with the perceptual illusions, as shown in figures 8g and 8h. Thus, there is evidence to assume that the model is consistent with the perceptual mechanisms. Following this conclusion and knowing that the PCNN has been used for object segmentation and that the RF-PCNN develops perceptual mechanism similar to the neurocomputational model PGLISSOM[15], the RF-PCNN model is proposed as the basis for the method of texture and shape segmentation. Thereby, V1/V2 extracts the features of texture shape with the PCNN and color with the TRSOM. Both features should be combined in a method inspired by the operation of V4 for complete the static object segmentation.

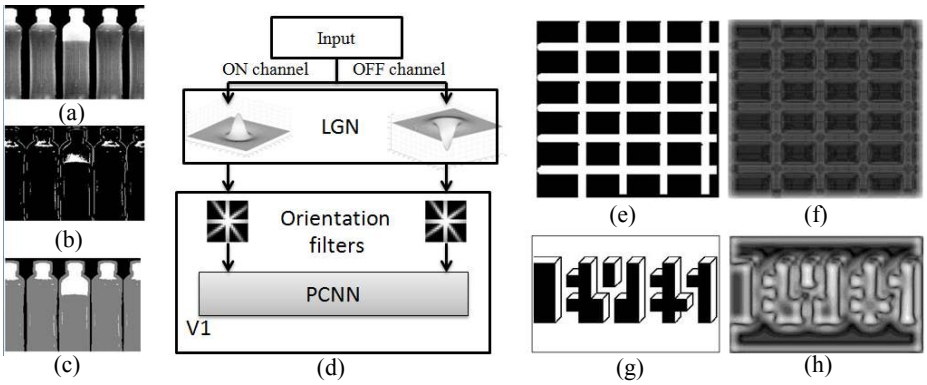


Fig. 8 PCNN model. (a). Input image for segmentation using PCNN time signatures. (b). Pulse in n=15. (c). Segmentation of (a) given by eq. (3). (d). RF-PCNN model. (e). Input based on the *Hermann*. (f). Output of the model RF-PCNN with *Hermann* illusion. (g). Image with hidden word '*eyes*'. (h). Output of the model RF-PCNN with hidden word 'eyes'.

4 Design of object segmentation methods

In the case of static object segmentation, we are developing a method inspired in V4, which is based on features of texture and color. The proposed scheme obtains the color feature from the segmentation generated by the TRSOM, and the texture features from the segmentation inspired by the RF-PCNN model. The inputs to this models are the V information defined as the maximum value in a RGB pixel and the neighborhood information of each pixel. The figures 9b and 9e shows the results of the both segmentation process.

Fig. 9. Color and texture analysis in video sequences. (a). Frame of the video'*14_p2007_3*'. (b). Color segmentation. (c). Frame of the video '*WS*'. (d). V information of (c). (e). Texture segmentation of (d).

With respect the dynamic object segmentation, we design a method based on background subtraction approach to extract all the motion regions, which is defined as visual alert. Also, with the aim to find the dynamic objects, we included an analysis of spatial and temporal illumination changes, which are defined by an accumulation of the differences between consecutive frames in temporal sequences. This accumulation forms a spatiotemporal pattern that describes the motion of dynamic objects. This spatiotemporal pattern represents the process in V1 to find motion patterns [3].

The background subtraction (BS) method is based on topographic interactions in V1, therefore we designed a model inspired in SOMRM for background modeling. The motion detection is based on BS and a model involving Cellular Neural Networks

to reduce noise. Both, background subtraction and the illumination changes are correlated to find the dynamic objects in a method inspired in the behavior of MT. Figure 10 shows a scheme of the proposed method for dynamic object segmentation along with a comparison of a segmentation and its groundtruth.

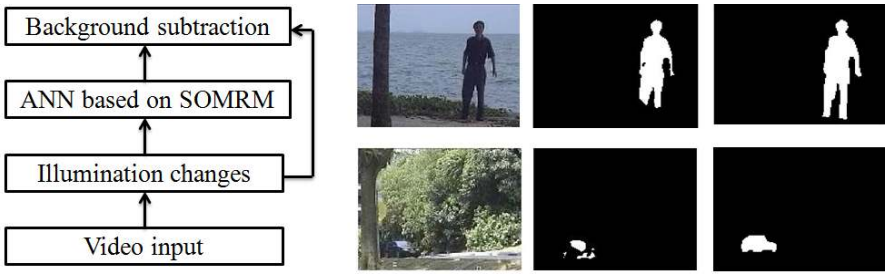


Fig. 10. Dynamic object segmentation method performs. (a). Scheme of the method. (b). Two video frames. (c). Dynamic object segmentation. (d) Groundtruth.

The validation of the dynamic object segmentation method was based on the metrics of Precision (P), Recall (R), F1and Similarity (S), all defined in [19], because these metrics are widely used in the literature related to dynamic object segmentation. The videos used were obtained from databases used in the literature and the methods were tested in critical condition. Table 1 presents the results which indicate a good segmentation performance except when dynamic objects have chromatic features similar to the background. The videos have a resolution of 120x160 and the dynamic object segmentation method run at 40 frames per second (fps). The videos were obtained from databases wallflowers and perception, which have been used to validate different dynamic segmentation methods.

Table 1. Dynamic Object Segmentation Results.

Video	CF	MR	FT	LB	MO	TD	WS	WT
P	0.9687	0.9254	0.875	0.9556	1	0.7876	0.9871	0.9769
R	0.9291	0.9177	0.7794	0.9188	1	0.7546	0.8328	0.9074
F1	0.9485	0.9215	0.8244	0.937	1	0.7707	0.9034	0.941
S	0.902	0.9215	0.7013	0.8811	1	0.627	0.824	0.889

5 Final Discussion

In this paper we have presented the progress in the implementation of a bioinspired method for static and dynamic object segmentation in video sequences, which is based on perceptual models of the visual cortex. This method was proposed by assuming that object detection is given by the analysis of features like color orientation and depth, while motion detection is based on spatiotemporal analysis of illumination changes and visual alert. From these models, we designed different methods to extract features of color, texture, alert and illumination changes, which were used to develop a set of methods that form a neuroinspired object segmentation scheme. Findings showed that the object segmentation is coherent except when the

color features of the objects are very similar between them. For this reason, future work we will be oriented to finish a method for static object segmentation inspired in V4 and we will develop a feedback scheme as in the visual cortex models to improve the perception process in the object segmentation.

6 References

1. Schwartz S.: Visual Perception a clinical orientation. McGraw Hill (2010) 1-3.
2. Cao Y. and Grossberg S.: Laminar Cortical Model of Stereopsis and 3D Surface Perception: Closure and da Vinci Stereopsis. Technical Report, (2004).
3. Berzhanskaya J., Grossberg S., Mingolla E.: Laminar cortical dynamics of visual form and motion interactions during coherent object motion perception. Technical Report, (2007).
4. Yu G. and Slotine J.: Visual Grouping by Neural Oscillator Networks, IEEE trans on Neural networks Vol. 20, No 12, pp 1871 – 1884, (2009).
5. Izhikevich E.: Which Model to Use for Cortical Spiking Neurons?, IEEE trans on Neural networks, vol 15, No 5, pp 1063-1070, (2004).
6. Benicasa A., Quiles M., Zhao L. and Romero R.: An Object-Based Visual Selection Model with Bottom-up and Top-down modulations; Brazilian Symposium on Neural Networks, pp 238-243, (2012).
7. Koene R., Hasselmo M.: An integrate and fire model of prefrontal cortex provides a biological implementation of action selection in reinforcement learning theory that reuses known representations, Int. Conf. on Neural networks, pp 2873-2878, (2005).
8. Ratnasi S. and McGinnity T.M.: A Spiking Neural Network for Tactile Form Based Object Recognition; Int. Join Conf on Neural Networks, pp 880-885, (2011).
9. Baier V.: Motion Perception with Recurrent Self-Organizing Maps Based Models, Int. Joint Conf. on Neural Networks, 1182-1186, (2005).
10. Kiang M.: Extending the Kohonen self-organizing map networks for clustering analysis, Elsevier Computational statistic & data analysis, Vol 38, No 2, pp 161-180, (2001).
11. Chacón-Murguia M. I. and Gonzalez-Duarte S.: An Adaptive Neural-Fuzzy Approach for Object Detection in Dynamic Backgrounds for Surveillance Systems, IEEE Trans. on Industrial electronics, Vol. 59, No. 8, , pp. 3286-3298, (2012).
12. 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 Trans. on Pattern Analysis and Machine Intelligence, Vol PP, No 99, 1-24, (2012).
13. Ben De Paula J.: Modeling the self-organization of color-selective neurons in the visual cortex, Report AI-TR-07-347, (2007).
14. Yu B., Zhang L.: Pulse-Coupled Neural Networks for Contour and Motion Matchings, IEEE trans on Neural networks, Vol. 15, No 5, pp 1186-101, (2004).
15. Miikkulainen R., Bednar J., Choe Y., Sirosh J.: Computational Maps in the Visual Cortex, Springer Sciences Media Inc, (2005).
16. Bednar J.: Building a mechanistic model of the development and function of the primary visual cortex, Journal of Physiology, pp 194-211, (2012).
17. Wang Z., Ma Y., Cheng F., Yang L.: Review of pulse-coupled neural networks, Image and Vision Computing, Vol 28, No 1, pp 5–13, (2010).
18. Frisby J. and Stone J., Seeing, The computational approach to biological vision, MIT Press, (2010).
19. Fan-Chieh C., Shih-Chia H. and Shanq-Jang R., "Illumination-Sensitive Background Modeling Approach for Accurate Moving Object Detection", IEEE Trans. on broadcasting, Vol 57, No 4, pp 794-801, (2011).

Wrapper method based on Soft Computing for Channel Selection in Brain Computer Interfaces

Alejandro Antonio Torres-García, Carlos Alberto Reyes-García, and Luis Villaseñor-Pineda

National Institute of Astrophysics, Optics and Electronics (INAOE).
Computer Science Department
Luis Enrique Erro # 1, Sta. María Tonantzintla, Puebla, Mexico.
{alejandro.torres,kargaxxi,villasen}@ccc.inaoep.mx

Abstract. Brain-computer interfaces (BCI) tries to provide to a subject, in specific to his/her brain, of a non-muscular channel to interact with electro-mechanic devices. In this work, we present a channel selection method to be applied in BCI based on electroencephalography (EEG). Specifically, BCIs whose electrophysiological sources are imagined speech (sometimes referred to as internal or unspoken speech) and motor imagery. In the first case, we used a dataset composed of EEG signals, belonging to twenty seven subjects, recorded during imagined speech with a protocol based on markers. Markers delimit the EEG signal segments of interest with aim to know a priori in what part a subject imagines the pronunciation of an specific word. Each segment of interest was filtered using common spatial reference (CAR). After that, discrete wavelet transform (DWT) and relative wavelet energy (RWE) were applied to extract features, and finally random forest (RF) was used to classify unspoken words. Our method outperforms previous works. However, in order to improve more the classification performance, we have observed the need to find a good combination of channels to identify better an specific unspoken word.

Key words: Electroencephalograms (EEG), Brain-Computer Interfaces (BCI), channel selection, imagined speech

1 Introduction

Electroencephalography (EEG) is a non-invasive, simple and relatively cheap technique to measure brain activity. These features are of special interest to the brain computer interface (BCI) research community. According to Wolpaw et al. [1], a BCI system tries to provide a new channel to the brain to transmit messages and commands to the external world. In general, BCI can be seen as a pattern recognition system where EEG is used as the primary source of raw information, and machine-learning algorithms are used to learn an inference function from EEG signals. The final goals of BCI research are to help handicapped persons and improve human-computer interaction (for example, in video game control).

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 53-62
Paper Received 20-03-2013 and Accepted 22-04-2013

Electrophysiological sources are the neurological mechanisms or processes employed by the BCI user to generate the control signals [2,1]. Among these, the most widely used are: slow cortical potentials (SCP), P300 potentials, motor imagery (sensorimotor rhythms mu and beta) and, visual evoked potentials (VEP). In recent years, unspoken speech (sometimes referred to as imagined speech) has been used as an electrophysiological source in BCI research (see the works described in [3,4,5,6]). According to Wester [7], unspoken speech is referred to as imagined pronunciation of a word without emitting a sound or articulating a facial movement.

Imaginary movements, like actual movements of different body parts, can produce attenuation of mu and beta rhythms at corresponding cortex locations. This circumscribed attenuation is called event-related desynchronization (ERD). Meanwhile, at other locations, an enhancement of both rhythms can be observed, called event-related synchronization (ERS) [8,9,10]. Briefly, different body parts are related to different locations in the motor and somatosensori cortex [11]. For example, movement or imagened movement of the left hand will cause ERD in the right motor cortex and ERS in the left motor cortex, and vice versa (see Figure 1) [12]. In general, the mu rhythm has a frequency band of 8–12 Hz and beta rhythm 18–25 Hz, but these frequency bands can vary depending upon the subjects and their mental states [13,14].

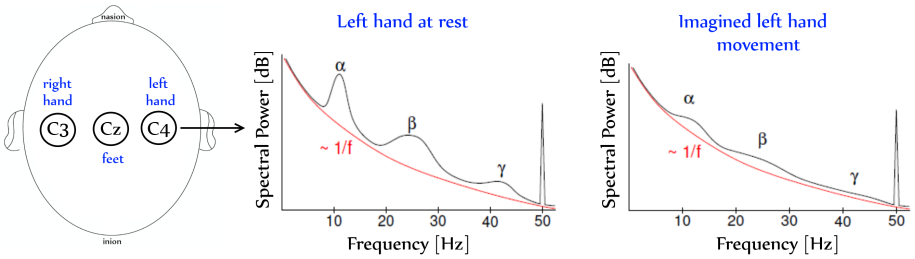


Fig. 1. Evidence of event-related desynchronization (ERD) and event-related synchronization (ERS) phenomena at channel C4 before and after onset imagined movement from the left hand (adapted from [15])

Our research is focused on motor imagery and unspoken speech because currently, BCIs based on motor imagery are the most widely used and unspoken speech is a relatively novel and interesting electrophysiological source to help prove a more natural communication channel without the need for translation [2]. BCI based on motor imagery is an independent system with high accuracy, and with the binary classification average of motor imagery tasks of the right and left hands above 90% [16].

However, both electrophysiological sources are not used in real-life applications. An important issue is that several algorithms available are focused on analyzing and processing information of multi-channel EEG. In BCI based on mo-

tor imagery, two approaches have been explored to tackle this problem: feature selection/dimensionality reduction and channel selection. In the first approach, principal component analysis (PCA) and independent component analysis (ICA) have been used. The second approach is relatively recent and looks for a more interpretable form to reduce the number of needed channels to achieve the same accuracy as a full channel configuration. The main difference between channel selection and feature selection is that information belonging to a channel is treated as a unique entity, with the main advantage that it results in a more interpretable dataset than feature selection. As with feature selection, channel selection can be divided in filter and wrapper methods, to our knowledge embedded methods have not been explored.

2 Previous works

To the best of our knowledge, an automatic channel selection method to unspoken speech has not been presented. Furthermore, at this time it is not clear what channels and brain regions are the most relevant for recognizing unspoken words from EEG signals. This last point has motivated the present research.

On the other hand, channel selection methods to motor imagery have been explored in several works, described as follows in the next subsections.

2.1 Channel selection based on filter methods

Lal et al. [17] adapted a feature extraction method known as recursive feature elimination to apply it in the channel selection context. Lal et al. [17] concluded that the number of channels used for classification can be reduced significantly without increasing the classification error. Another filter method was proposed in [18]. This work used mutual information (MI) maximization. EEG channels were ranked based on MI between the selected channel and a class label. Finally, Wang et al. [16] proposed a novel approach to select channels by common spatial patterns (CSP). Their work was applied to binary classification and their channel selection criterion was to use the first and the last columns in the resultant matrix of spatial patterns.

2.2 Channel selection based on wrapper methods

In [19], a channel selection method was proposed based on genetic algorithms and artificial neural networks (ANN). This work was applied to a binary classification problem, with three layers, and assessed by accuracy. Another work, described in [20], proposed a channel selection method based on genetic algorithms and linear support vector machine (LSVM). This method was only assessed by accuracy and was applied in a binary classification problem. Finally, [21] presented a method in which Binary Particle Swarm Optimization (BPSO) and CSP were utilized. This method aggregates a trade-off coefficient to modify the objective function. This coefficient was varied and the accuracy evaluated for each case.

The last works in channel selection based on wrapper methods have considered the possibility of simultaneously optimizing the number of selected channels and accuracy. In this case the optimization problem is multi-objective. Examples of kinds of methods were presented in [22,23]. Both methods were proposed for the same research group and were used to select channels in binary classification problems. The objectives were accuracy and number of selected channels.

2.3 Discussion

In our research, wrapper methods are of special interest because they are more accurate than filters. This is possible because wrappers use the same machine learning to assess the selected channels and to classify in the classification stage. The first works on wrapper methods only used accuracy as an objective. However, in real-life applications it is more important to process and analyze fewer channels. To our knowledge, previous works in multi-objective optimization have not attacked multiclass problems and they considered channels with artifacts in their space search. The works described in [21,16] are limited for binary nature of CSP method. These methods cannot be extended to multiclass problems. Furthermore, our work will use the same objectives proposed by Hasan et al. [23]: number of selected channels and error rate.

3 Research objective

To develop a channel selection method based on evolutionary multi-objective optimization whose accuracy and the number of selected channels will be comparable to previous works. This method will be robust against multiclass problems.

4 Methodology

The methodology is composed of the following stages: Collect data for experiments, pre-processing, feature extraction, artifacts removal, channel selection, design a classification model, and evaluation.

- **Collect data for experiments.** To realize our experiments we have collected BCI Competition Datasets. These datasets are characterized for multiclass and binary problems. Furthermore, we will use EEG datasets characterized for unspoken speech that were recorded in [24].
- **Pre-processing.** This stage will search to prepare the signals, improve the signal/noise ratio and, reject frequencies related with electromiographic signals.
- **Artifacts removal.** This stage searches to remove channels affected by artifacts such as blinking eyes, muscular movements and heartbeats. This part is very important because it reduces the search space for channel-selection stage.

- **Design classification model.** In this stage a classification strategy or model will be designed to use with multiclass problems.
- **Channel selection.** This stage will use a wrapper method and will explore different soft computing techniques (such as particle swarm optimization, ant colony optimization, genetic algorithms, memetic algorithms, among others) to experiment with on a multi-objective optimization problem and identify the best one for channel selection.
- **Evaluation.** In this stage, we will conduct statistical tests to compare the results of our method with previous works. The measures to compare are: accuracy and number of selected channels.

5 Experiments and preliminary results

We have analyzed datasets from unspoken speech used in [24]. This dataset is composed of EEG signals belonging to 27 native-Spanish-speaking subjects. EEG signals were recorded with an EMOTIV acquisition kit which has 14 channels and two reference electrodes (channel names in the 10-20 International System: AF3, AF4, F3, F4, F7, F8, FC5, FC6,P7, P8, T7, T8, O1, O2. References: P3/CMS, P4/DRL). This dataset was recorded using a basic protocol to acquire EEG signals from each subject. This protocol consists in placing a subject restfully sit with open eyes and with his/her right hand over a computer mouse. Subject delimits, by clicking a mouse to mark the EEG signals, both the start and the end of the imagined pronunciation of each of the words belonging to a reduced vocabulary composed of five Spanish words: "arriba," "abajo," "izquierda," "derecha," and "seleccionar". The aim behind of the adquisition protocol is to know a priori in what part of the EEG signal, it is necessary to search the associated patterns with the imagined pronunciation of an indicated word.

The interest segments of the EEG signals are those between the start and end markers, these segments are called epochs. Each epoch has variable duration like in spoken speech. Furthermore, each word was internally pronounced thirty three times consecutively. Before this, it was indicated to the subject what word had to internally pronounce. All blocks of words belonging to a same subject were recorded in a single session (same day). All sessions were recorded in a laboratory far from audible and visual noise.

It is important to mentione that before that the EEG signal recording is started, it was indicated to the subject to avoid blinking and any corporal movements while imagining the pronunciation of the word in turn. After each end marker the subject can take a short break to do these movements in case he/she needs to move. Moreover, to avoid the subjects distraction by counting the repetitions of imagined words or how many of them were still left, they did not know how many times the words would be repeated. For this, a control assistant, inside the recording room, internally counts the repetitions and indicates when the session should be concluded.

On the other hand, in [24], the researchers processed a subset of four EEG channels (F7-FC5-P7-T7) with their method. They assumed this combination

is a good selection because these channels are the nearest to the Broca and Wernicke brain areas. But they did not explore other combinations. More recent works have suggested, however, that other brain regions can be involved with imagined speech [25]. Therefore, it is interesting to study whether EEG channels over these brain regions can be selected using a search process.

We have designed a novel method to process EEG signals. This method applies common average reference (CAR) over the EEG channels previously delimited with markers. With this, the average voltage of each sample in time from all channels is subtracted of each channel. After that, to each marked segment of each of the channels, discrete wavelet transform (DWT) is applied using a Daubechies-2 as the mother wavelet and with five decomposition levels (D1-D5 and A5 levels). Then, with each obtained DWT, relative wavelet energy (RWE) is computed to normalize the coefficients. With this, each marked segment of each EEG channel is represented with 6 coefficients, but the first coefficient corresponding with the frequencies between 32-64 is discarded. With this, each marked segment of each EEG channel is represented with the five coefficients remaining (RWE from D2-D5 and A5 levels). After that, simultaneous coefficients were concatenated to form a feature vector. Last, a random forest (RF) is trained with many feature vectors to classify each EEG marked segment in its corresponding class (any of five Spanish words). Figure 2 shows the novel method used to process EEG signals.

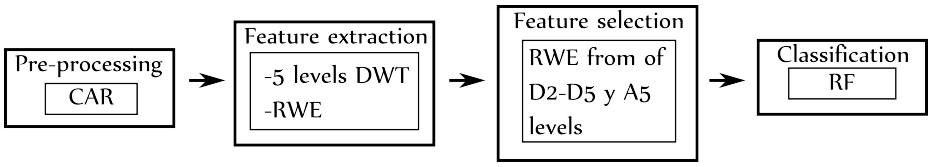


Fig. 2. Method used to process EEG signals recorded while a subject imagines word pronunciation

5.1 Experiments and results using the nearest four channels to the brain's linguistic areas

Our first experiment consisted of assessing the accuracy of our method compared with the method used in [24]. For this, we processed only four EEG channels of each subject. The channels are F7, FC5, P7 and T7, i.e., the same channels used in [24]. Figure 3 shows the percent of accuracy obtained by both methods, applying 10-fold cross validation, and using only four channels. In general, our method outperformed to the method used in [24]. Subjects S5, S7, S11, S13, S18 and S22 were not processed in [24] due to assumptions on their method as: subjects left-handed (S13 and S18) have their lenguage areas on right hemisphere of the brain (the four channels are not on this brain side), and subjects (S5, S7,

S11 and S22) with many EEG marked segments whose size is more than 256 time samples were discarded.

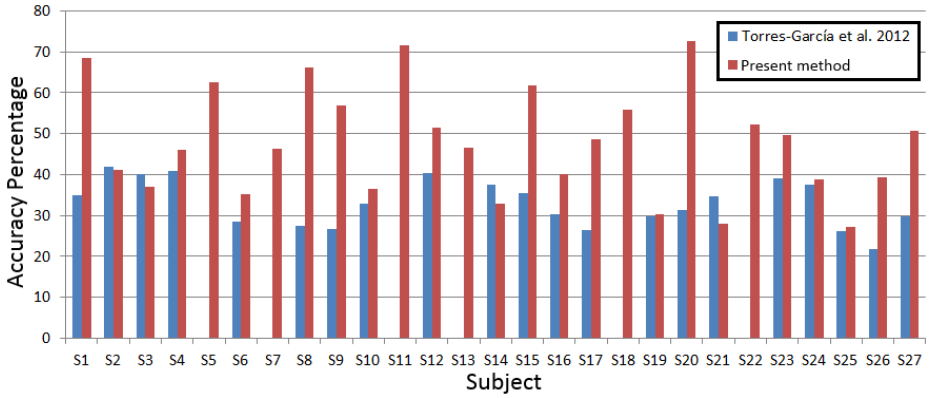


Fig. 3. Accuracy percentages obtained by both Torres-García et al. [24] and the present methods. Remarking that subjects S5, S7, S11, S13, S18 and S22 were not processed in [24]

5.2 Experiments and results using all channels

The second experiment consisted of assessing whether the use of others channels, out of the brain's linguistic areas, could contribute to improve the accuracy obtained using four channels (F7-FC5-T7-P7). At the same time, this could give evidence about how good it was the idea to use them taking into account that those channels are the nearest to the Broca and Wernicke areas. To this, the EEG signals were processed with our method, in which the classifier was trained and tested using the EEG signals with two distinct channels configurations: four (F7-FC5-T7-P7) and all (fourteen) channels. The results can be seen in Table 1. In general, the accuracies obtained using 14 channels are better than using four. It is important to mention that even though four channels (F7-FC5-T7-P7) result in a contribution coefficient greater than 0.5 for all subjects (that is, accuracy of four channels was 50-percent or more of 14 channels), this is not sufficient to conclude that only these channels contain information associated with the imagined pronunciation of words. These results are in sintony with the works mentioning that other brain regions can be involved in the imagined speech process. Furthermore, these results motivate the search for a more accurate minimal subset of channels than the 14-channels configuration.

Table 1. Accuracy percentages obtained using 4 channels and 14 channels. Column "contribution coefficient" is achieved by dividing the results of using 4 channels into the results achieved with 14 channels

	accuracy percentage		contribution
Subject	4 channels	14 channels	coefficient
S1	68.48	**80**	0.86
S2	41.21	**49.09**	0.84
S3	36.96	**63.03**	0.59
S4	46.06	**58.18**	0.79
S5	62.42	**67.87**	0.92
S6	35.15	**43.03**	0.82
S7	46.34	**64.02**	0.72
S8	66.06	**86.06**	0.77
S9	56.96	**62.42**	0.91
S10	36.36	**61.21**	0.59
S11	71.51	**83.63**	0.86
S12	51.51	**60.6**	0.85
S13	46.66	**65.45**	0.71
S14	32.72	**43.03**	0.76
S15	**61.81**	60.6	1.02
S16	40	**50.9**	0.79
S17	48.48	**67.27**	0.72
S18	55.75	**72.12**	0.77
S19	30.3	**51.51**	0.59
S20	72.72	**76.36**	0.95
S21	27.87	**36.96**	0.75
S22	52.12	**65.45**	0.8
S23	49.69	**53.93**	0.92
S24	38.78	**45.45**	0.85
S25	27.27	**43.63**	0.63
S26	39.39	**52.72**	0.75
S27	50.6	**59.14**	0.86

6 State of the research

We have studied the present EEG channel selection methods to delimit the scope of our research. Furthermore, we will begin to explore feature extraction methods, specifically, discrete wavelet transform and relative wavelet energy. We have defined two objectives for minimizing the number of selected channels and error rate (complement of accuracy).

7 Conclusions

We have proposed a novel approach for channel selection that uses a preliminary reduction of the number of channels by removing channels containing artifacts.

Furthermore, this allows a reduction in search space. The multi-objective optimization will help to find better solutions than mono-objective optimization because accuracy is not an exclusive measure in real-life applications.

On the other hand, we have proven the importance of channel selection in unspoken speech datasets. Our research only compares four and 14 channels, with 14 proving best. However, processing 14 channels is more costly than using fewer channels. The next step in our research is to determine which, and how many, channels are the best configuration for this task.

References

1. J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. Vaughan, "Brain-computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.
2. J. Brumberg, A. Nieto-Castanon, P. Kennedy, and F. Guenther, "Brain-computer interfaces for speech communication," *Speech communication*, vol. 52, no. 4, pp. 367–379, 2010.
3. K. Brigham and B. Kumar, "Imagined Speech Classification with EEG Signals for Silent Communication: A Preliminary Investigation into Synthetic Telepathy," in *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference on*, pp. 1–4, IEEE, 2010.
4. A. Porbadnigk, "EEG-based Speech Recognition: Impact of Experimental Design on Performance," Master's thesis, Institut für Theoretische Informatik Universität Karlsruhe (TH), Karlsruhe, Germany, 2008.
5. M. D'Zmura, S. Deng, T. Lappas, S. Thorpe, and R. Srinivasan, "Toward EEG sensing of imagined speech," *Human-Computer Interaction. New Trends*, pp. 40–48, 2009.
6. C. DaSalla, H. Kambara, M. Sato, and Y. Koike, "Single-trial classification of vowel speech imagery using common spatial patterns," *Neural Networks*, vol. 22, no. 9, pp. 1334–1339, 2009.
7. M. Wester, "Unspoken speech-speech recognition based on electroencephalography," Master's thesis, Universität Karlsruhe (TH), Karlsruhe, Germany, 2006.
8. G. Pfurtscheller and A. Aranibar, "Evaluation of event-related desynchronization (erd) preceding and following voluntary self-paced movement," *Electroencephalography and clinical neurophysiology*, vol. 46, no. 2, pp. 138–146, 1979.
9. G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, "Eeg-based discrimination between imagination of right and left hand movement," *Electroencephalography and clinical Neurophysiology*, vol. 103, no. 6, pp. 642–651, 1997.
10. G. Pfurtscheller, C. Brunner, A. Schlögl, S. Lopes, *et al.*, "Mu rhythm (de) synchronization and eeg single-trial classification of different motor imagery tasks.," *NeuroImage*, vol. 31, no. 1, p. 153, 2006.
11. B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. Muller, "Optimizing spatial filters for robust eeg single-trial analysis," *Signal Processing Magazine, IEEE*, vol. 25, no. 1, pp. 41–56, 2008.
12. G. Liu, G. Huang, J. Meng, and X. Zhu, "A frequency-weighted method combined with common spatial patterns for electroencephalogram classification in brain–computer interface," *Biomedical Signal Processing and Control*, vol. 5, no. 2, pp. 174–180, 2010.

13. C. Andrew and G. Pfurtscheller, "On the existence of different alpha band rhythms in the hand area of man," *Neuroscience letters*, vol. 222, no. 2, pp. 103–106, 1997.
14. G. Pfurtscheller, A. Stancak, and G. Edlinger, "On the existence of different types of central beta rhythms below 30 hz," *Electroencephalography and clinical neurophysiology*, vol. 102, no. 4, pp. 316–325, 1997.
15. K.-R. Müller and B. Blankertz, "Machine Learning and Signal Processing Tools for BCI." `http://videolectures.net/site/normal_dl/tag=46578/bbci09_blankertz_muller_mlasp_01.pdf`, 2009.
16. Y. Wang, S. Gao, and X. Gao, "Common spatial pattern method for channel selelction in motor imagery based brain-computer interface," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 5392–5395, IEEE, 2006.
17. T. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf, "Support vector channel selection in bci," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 6, pp. 1003–1010, 2004.
18. T. Lan, D. Erdogmus, A. Adami, M. Pavel, and S. Mathan, "Salient eeg channel selection in brain computer interfaces by mutual information maximization," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 7064–7067, IEEE, 2006.
19. J. Yang, H. Singh, E. Hines, F. Schlaghecken, D. Iliescu, M. Leeson, and N. Stocks, "Channel selection and classification of electroencephalogram signals: An artificial neural network and genetic algorithm-based approach," *Artificial intelligence in medicine*, 2012.
20. A. Al-Ani and A. Al-Sukker, "Effect of feature and channel selection on eeg classification," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pp. 2171–2174, IEEE, 2006.
21. J. Lv and M. Liu, "Common spatial pattern and particle swarm optimization for channel selection in bci," in *Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on*, pp. 457–457, IEEE, 2008.
22. N. Al Moubayed, B. Hasan, J. Gan, A. Petrovski, and J. McCall, "Binary-sdmopso and its application in channel selection for brain-computer interfaces," in *Computational Intelligence (UKCI), 2010 UK Workshop on*, pp. 1–6, IEEE, 2010.
23. B. Hasan and J. Gan, "Multi-objective particle swarm optimization for channel selection in brain-computer interfaces," in *The UK Workshop on Computational Intelligence (UKCI2009)*, 2009.
24. A. A. Torres-García, C. A. Reyes-García, and L. Villaseñor-Pineda, "Toward a silent speech interface based on unspoken speech," in *BIOSTEC - BIOSIGNALS* (S. V. Huffel, C. M. B. A. Correia, A. L. N. Fred, and H. Gamboa, eds.), pp. 370–373, SciTePress, 2012.
25. J. Binder, J. Frost, T. Hammeke, R. Cox, S. Rao, and T. Prieto, "Human brain language areas identified by functional magnetic resonance imaging," *The Journal of Neuroscience*, vol. 17, no. 1, pp. 353–362, 1997.

Methodology for automatic evaluation of restricted domain ontologies *

Mireya Tovar[1,2], Azucena Montes[1,3], and David Pinto[2]

[1]Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Mexico
[2] Faculty Computer Science, Benemérita Universidad Autónoma de Puebla, Mexico,
[3]Engineering Institute, Universidad Nacional Autónoma de Mexico.
{mtovar,amontes}@cenidet.edu.mx
{dpinto,mtovar}@cs.buap.mx

Abstract. In this paper we present advances of the PhD research thesis entitled: "Automatic Evaluation of Restricted-Domain Ontologies". We discuss the methodology employed, so as the results obtained up to now. The evaluation has been carried out over two main components of the ontology: concepts and relationships. Thus, on the one hand we present methods for discovering and validating concepts. On the other hand, we show results when the degree of semantic similarity among concepts is computed over the relationships that already occur in the ontology to be evaluated.

Key words: Ontology evaluation, Natural Language Processing, lexical-syntactic patterns.

1 Introduction

In recent years, especially with the emergence of the concept of the Semantic Web, it is understood that there is a great interest in the management of ontological resources with the aim of help to comply the user information needs. The World Wide Web is one of the largest public repositories of information, but most of that information is designed for human consumption, thereby, it is almost innaccessible for machines because it is not properly organized, thus making difficult for software applications to use the WWW in an automatic way [1]. When semantic is used to organize or structure the information at the Web, this data receives the name of Semantic Web. Organization of knowledge in the Semantic Web is usually performed by means of ontologies. An ontology uses a predefined, reserved vocabulary of terms to define concepts and the relationships between them for one specific area of interest, or domain [1]. Gruber [2] defines an ontology as: "an explicit specification of a conceptualiation".

An ontology includes classes, instances, attributes, relationships, constraints, rules, events and axioms. In many cases, ontologies are structured as hierarchies

* This work is partially supported by CONACYT and PROMEP under grants: CONACYT 54371, PROMEP/103.5/12/4962 BUAP-792 and project CONACYT 106625.

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 63-72
Paper Received 28-02-2013 and Accepted 22-04-2013

of concepts modeled either by means of part-whole or class-inclusion semantic relationships. There exist, other types of semantic relationships that are not hierarchical such as synonyms, antonyms, etc.

Although there is sufficient research on methodologies, techniques, tools and software for building ontologies, an aspect that has not been considered in depth is the evaluation of ontologies. One of the reasons of this phenomenon is the difficulty for determining which items should be evaluated and what criteria should be considered to specify the quality of the ontology [3]. In this research work we aim to develop an automatic method for evaluating restricted-domain ontologies, employing Natural Language Processing (NLP) techniques. The methodology proposed assumes that one ontology has been automatically, semi-automatically or manually constructed. Therefore, we aim to "validate" the quality of the elements of the ontology, such as relationships and concepts. The evaluation is carried out twofold, 1) By using a reference corpus (document collection) of the same ontology domain, and, 2) by using self-evaluation, considering only the information stored in the ontology itself. Eventhough, two evaluations methods are considered in the PhD thesis, in this report, we only present results with respect to the first proposal.

The remaining of this paper is structured as follows. Section 2 describes with more detail the problem we are dealing with. The methodology proposed for solving this problem is presented in Section 3. A description of the contribution expected in this research work is presented in Section 4. The results obtained up to now for the evaluation of concepts and relationships is given in Section 5. Finally, in Section 6, the conclusions of the advances of this PhD thesis are given.

2 Research problem to solve

As previously stated, the problem we are intended to deal with is the automatic evaluation ontologies. We assume the concept of evaluation, in our context, to be associated with the process of determining the quality of such ontologies. According to literature, evaluation of ontologies can be performed in one or two of the following stages: a) During the construction of the ontology, and b) Once the ontology has been constructed.

If the ontology is small (for instance, with small number of concepts and relationships), the first approach is practical for verifying the quality of it. However, when the ontology contains a large number of components, the time it would take for a human expert for evaluating it could be very expensive in terms of time. Thus, the proposal of novels methods for automatic evaluation of ontologies (independently of the construction stage) would be of high benefit. The aim is to provide a framework which allows an extra tool to the ontological engineer for constructing better knowledge databases than when none evaluation tools is used.

There are various ontology evaluation approaches reportes in literature. Some of these approach descriptions follows: 1) Based on criteria ([4], [5], [6], [7], [8]);

2) Based on gold standard ([9], [10], [11]), corpus-based or data ([3], [12], [13], [14]); 3) Based in tasks ([15]) or application ([16]).

In particular, our proposal intend to use the criteria and corpus based approaches. One of the criteria we consider very important for the development of this research work is the correctness criterion. According to [5], **Correctness** specifies whether or not the information stored in the ontology is true, independently of the domain of interest.

In summary, we propose the automatic evaluation of domain ontologies, considering content level evaluation in the domain corpus, based in the correctness criterio, i.e., to verify whether or not the concepts and relationships of a domain ontology are true. For this purpose, we use boolean scores: 1 if the concept set or relationship set is correct in the domain corpus, or 0 otherwise. Additionally, we evaluate the semantic similarity degree of the ontology relationships using score values normalized between 0 and 1, such as Jaccard coefficient, Dice, etc.

3 Methodology

The methodological solution for this research works considers evaluate any type of domain ontology, assuming that exists a reference corpus. One of the initial conditions for the evaluation of a given ontology is that is should be well designed, structurally speaking, and that the reference corpus corresponds to the same domain of the ontology. Even if, this corpus can contain any domain-specific texts, i. e., scientific publications, project reports, books, medical notes, etc., it is expected that it sufficient diverse, i.e., to guarantee that there exist a reasonable amount of text [17] for executing statistical methods for searching evidence of the ontology components to be evaluated.

The proposed methodology for evaluation ontologies proposes three phases: a) Information filtering, b) Discovery of candidate terms and candidate relationships, and c) Evaluation of the ontology components, which are described in the following sub-sections.

3.1 Information filtering

Even if, there exist a reference corpus, it is normally made up of a huge amount of documents which need to be filtered in order to obtain the most specific features to be used in the process of evaluating the components of the ontology. The proposed method considers to find the most suitable information for validating each triple of the ontology. In this sense, the architecture considers the following three sub-modules:

Extraction of triples In this sub-module, we consider the theory of OWL[1], and we implement two algorithms in Jena[2] for extracting classes (or concepts) and relationships of restricted-domain ontologies.

[1] `http://www.w3.org/TR/owl-features/`
[2] `http://jena.apache.org/`

Query construction From the information extracted from the ontology, we construct queries that will be used for searching evidence in the reference corpus. This is perhaps one of the most critical steps of the methodology. Finding evidence of the triple quality in the reference corpus is actually the aim of this research work. Given a triple (S, R, O), with S the subject, R the relationship and O the object, the queries are constructed as: $S * O$ (documents containing S near of O), $*S*$ (documents containing S) and $*O*$ (documents containing O).

Information Retrieval System The third sub-module uses de queries constructed as input in an information retrieval system for finding information associated to the triples. The purpose of this phase is to filter out documents of the domain corpus containing terms of the ontology.

3.2 Discovery of candidate terms and relationships

The aim of this PhD thesis is to evaluate the quality of the ontology components. Thus, we might find or discover terms and relationships in the reference corpus in order to "validate" the terms and relationships already stored in the ontology. This process of discovering is explained as follows.

Discovery of candidate terms Up to now, we use a pattern-based approach for discovering candidate terms in the reference corpus, i.e. we find those terms that match one or more of the lexical-syntactic patterns already discovered in the ontology.

For discovering the candidate terms in the reference corpus, which may or not correspond to the concepts of the domain ontology, we implemented the following procedure:

1. To apply a Part-of-Speech (PoS) tagger (Tree Tagger[3]) to the concepts of the ontology to be evaluated [18].
2. To cluster similar concepts by using the morphological PoS tags.
3. To create morphological and/or lexical-syntactic patterns from the concept clusters.
4. To use the lexical-syntactic patters in order to extract candidate terms from the reference corpus.

The above procedure produces a list of candidate terms that will be further used for determining whether or not a concept should be present in the ontology, based on the evidence that exist in the reference corpus.

Discovery of candidate relationships We have also planned to identify lexical-syntactic patterns for identifying relationships of the ontology to be evaluated. For this purpose, we also need to find evidence in the reference corpus

[3] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

about a given ontology relationship. We have considered to use a hybrid approach for the discoverying of the candidate relationships in the corpus, considering a pattern based approach together with a pure statistical approach. The former approach has not beed developed yet, however, the latter is considered in this paper. We use similarity measures for determining the degree of association between a pair of concepts. In other words, we are now only evaluating the quality of the relationship by measuring how related these two concepts are.

For this purpose, we identify the concepts associated to the relationship of the ontology and we construct frecuency vectors using the vocabulary of the domain corpus for each concept. We apply the similarity measure directly to the vectors and we determine the degree of term correlation associated to each relationship of the ontology. The hypothesis follows: "the most similar the concept vectors are, the better the quality of the relationship".

3.3 Evaluation measures

The third phase of the solution architecture is based on the evaluation of ontology triplets, using candidate terms and relationships generated in the automatic discovery phase.

Concept evaluation measures We have evaluated the performance of the presented approach by means of standard evaluation measures such as precision, recall and F-measure. The precision is the proportion of the candidate terms which truly are concepts in the ontology among all those which were identificated as candidate terms. The recall is the proportion of candidate terms which were identificated as concepts in the ontology, among all the real concepts of the ontology. The F-measure is the harmonic mean of precision and recall. These measures are defined as follows:

$$Precision = \frac{CO\_CexR}{reExCor} \tag{1}$$

$$Recall = \frac{CO\_CexR}{COnt} \tag{2}$$

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

Where:

CO_CexR	Number of multi-word terms extracted by the lexical-syntactic patterns that overlap between the ontology and the corpus.
$reExCor$	Total number of candidate terms extracted from the corpus using the lexical-syntactic patterns.
$COnt$	Total number of concepts of the ontology.

Evaluation of semantic relationships The similarity measures considered so far for evaluating the degree of relationship for a pair of concepts (x, y) are: cosine ($d(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^t\boldsymbol{y}}{\|x\|\|y\|}$), Tanimoto or Jaccard ($d(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^t\boldsymbol{y}}{\boldsymbol{x}^t\boldsymbol{x}+\boldsymbol{y}^t\boldsymbol{y}-\boldsymbol{x}^t\boldsymbol{y}}$), Sockal ($d(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^t\boldsymbol{y}}{2\boldsymbol{x}^t\boldsymbol{x}+2\boldsymbol{y}^t\boldsymbol{y}-3\boldsymbol{x}^t\boldsymbol{y}}$), dice coefficient ($d(\boldsymbol{x}, \boldsymbol{y}) = 2\frac{\boldsymbol{x}^t\boldsymbol{y}}{\boldsymbol{x}^t\boldsymbol{x}+\boldsymbol{y}^t\boldsymbol{y}}$), and a variation of the previous measures (we have named it SimVar- $d(\boldsymbol{x}, \boldsymbol{y}) = 3\frac{\boldsymbol{x}^t\boldsymbol{y}}{\boldsymbol{x}^t\boldsymbol{x}+\boldsymbol{y}^t\boldsymbol{y}+\boldsymbol{x}^t\boldsymbol{y}}$) [19]. In order to be consistent in terms of the average results, we have reported random variables instead of similarity values.

Thus, the evaluation measures are summarized in terms of the following random variables: Mean ($\bar{\mu}$), Standard deviation (σ) and Coefficients of variation ($CV = \frac{\sigma}{\bar{\mu}}$).

Standard deviation shows the variation or dispersion of the data with respect to the mean. A low standard deviation indicates that the data tend to be close to the mean, a high standard deviation indicates that the data has high variation or dispersion. It is expressed in the same units as the data. To compare results between different data sets, regardless of the units, the coefficients of variation are used.

4 Main contribution

This proposal aims to evaluate the ontology independently of the construction phase, i. e. when the ontology is in development or when it is already constructed. Thus, providing the end user or the engineer for an ontological evaluation of the ontology in those cases when the knowledge database is too large. In this research proposal we have considered only restricted-domain ontologies, assuming that there exist a reference corpus for the evaluation process.

The main contribution of this proposal is the introduction of automatic methods for validating ontologies automatically, semi-automatically or even manually constructed. We provide mechanisms for discovering candidate terms and relationships from a reference corpus which may be further used for validating those terms and relationships that already exist in the ontology to be evaluated.

5 Results achieved and their validity

In this section we present the results obtained with the proposed approach.

5.1 Dataset

Table 1 presents two characteristics of the two ontologies already evaluated in this research paper. The number of concepts and "hierarchical" [4] relationships of the two ontologies.

[4] In this paper we have considered only those relationships extracted by means of the "subClassOf" axiom of OWL.

Table 1. Domain ontologies.

Domain	owl file	Total of concepts	Hierarchical relationships
Petroleum	petroleum.owl	48	37
Artificial Intelligence	ai.owl	276	205

Table 2 shows the number of documents processed from the reference corpus, together with the total of tokens analized. In that table, it can be seen a new corpus which results from filter the petroleum corpus by using the information filtering techniques described above.

In the case of the petroleum domain, the number of documents containing the terms of the concepts of the domain ontology are 575. For the case of Artificial Intelligence corpus, the filtering step resulted in a subcorpus exactly equal to the total of documents of the corpus, thus we have not added another row.

Table 2. Corpora to be evaluated.

Corpus	Documents	Tokens
Petroleum domain	577	9,730,495
Petroleum Filtered Subcorpus	575	9,727,092
Artificial Intelligence	8	10,805

5.2 Results for the evaluation of concepts

Tables 3 and 4 show the lexical-syntactic patterns obtained by the procedure presented in Section 3.2, and the results obtained by applying these patterns to the reference corpus for the Petroleum and Artificial Intelligence domain, respectively. The first column indicates the number of patterns identified in the ontology, column two indicates the frequency of the pattern, column three is the pattern identified in the ontology, finally, the last column shows the sum of term frequencies availables in the corpus. Only those terms that fulfill with the corresponding lexical-syntactic pattern. The last row indicates the number of candidate terms extracted from the reference corpus without repetition.

Consider the following tags for the petroleum ontology: IN is a preposition, RB is an adverb, NN is a Noun, JJ is an adjetive, VB is a verb in participle past, or verb gerund (VBP or VBG). For the Artificial Intelligence ontology, the tags follows: NN is a noun, proper noun, plural noun (NN, NP or NNS), JJ is an adjective or superlative adjective (JJ or JJS), CD is a cardinal number, IN is a preposition, VB is a verb in any form (VBG, VBZ, VB, VBN, VBP or VBD), FW is a foreign word and RB is an adverb.

Once the candidate terms were extracted from the reference corpus using the lexical-syntactic patterns, we match them with respect to the concepts of the ontology. We noted that from the 48 concepts defined in the Petroleum ontology,

Table 3. Lexical-syntactic patterns for the extraction of candidate terms in the Petroleum domain corpus.

n	Fr Ont	Lexical-syntactic patterns	Fr corpus
1	21	NN^+ $JJ?$	1,823,294
2	11	NN $VB(JJ\|(NN^+)?)$	125,308
3	11	JJ $(NN^+)?$	646,029
4	5	RB $(VB?\|NN?)$	223,301
5	4	VB $(NN\|JJ)$	71,358
6	1	IN NN^+	192,879
		Total of unrepeated terms:	378,465

Table 4. Lexical-syntactic patterns for the extraction of candidate terms in the Artificial Intelligence domain corpus.

n	Fr Ont	Lexical-syntactic patterns	Fr corpus
1	243	$(NN^+)((VB\ NN)?\|(VB^+)?)$	2693
2	84	$(JJ)^+(NN^+)?$	1000
3	35	$NN\ ((JJ\ NN)\|(IN)(JJ)(NN)\|(CD)\|(IN\ VB)\|(VB\ JJ\ NN)\|(IN\ NN^+))$	400
4	17	$(VB^+)(NN^+)?$	1582
5	10	$(JJ\ NN)((IN\ JJ\ NN?)\|(VB\ NN?)\|(JJ\ NN))$	135
6	6	$JJ\ ((NN\ IN\ JJ\ NN?)\|(VB\ NN?))$	43
7	3	$RB\ ((VB\ JJ\ NN)\|(JJ\ NN))$	27
8	2	$(VB\ IN)(NN\|(JJ\ NN))$	105
9	1	$IN\ JJ\ NN$	151
10	1	$FW\ NN\ IN\ JJ\ NN$	0
		Total of unrepeated terms:	3592

the system was able to find 42 concepts. For the case of the Artificial Intelligence domain, the system found 205 of 276 concepts. The results of precision, recall and F-measure for the Petroleum and Artificial Intelligence domains are shown in Table 5.

Table 5. Results of evaluation measures applied to concepts.

Ontology	Recall	Precision	F-measure
Petroleum	0.875000	0.00011098	0.00022192
Artificial Intelligence	0.742754	0.05707130	0.10599799

5.3 Results for the evaluation of semantic relationships

The similarity measures presented in Section 3.2 were applied to the hierarchical relationships of the ontologies to be evaluated. Table 6 shows the results

obtained. For the Petroleum ontology (with 37 hierarchical relationships), the mean similarity measure were very close to 1.0, indicating a high correlationship between concepts sharing a relationship in the ontology. The $\bar{\mu}$ value indicates that exist low variation for each relationship identified. The CV value shows a 5% of dispersion among the five similarity measures. For the case of correctness, we use an score of 0.90 for indicating that the relationship is true in the domain corpus. We observed that we have obtained more than 31 relationships of the Petroleum ontology. In the Artificial Intelligence ontology, the $\bar{\mu}$ is quite variable for the five similarity measures, though we always obtained a value greater than 40% of similarity. In the case of σ and CV, we observed that the best result was when the coseno similarity was used. By using correctness (score of 0.90), we obtained 87 of 205 hierarchical relationships of the ontology.

Table 6. Results of the evaluation measures for the hierarchical relationships.

	Petroleum				Artificial Intelligence			
Similarity	$\bar{\mu}$	σ	CV	$Correctness$	$\bar{\mu}$	σ	CV	$Correctness$
Cosine	0.9994	0.000444	0.0004	37/37	0.8185	0.162945	0.199	87/205
Jaccard	0.9754	0.028671	0.0293	36/37	0.5835	0.310579	0.532	54/205
Sokal	0.9535	0.051713	0.0542	31/37	0.4841	0.335557	0.693	43/205
SimVar	0.9915	0.010339	0.0104	37/37	0.7405	0.244661	0.330	77/205
Dice	0.9874	0.015193	0.0153	37/37	0.6846	0.272201	0.397	64/205

6 Conclusions

In this paper we present advances of the PhD thesis that tackles the problem of automatic evaluation of restricted-domain ontologies. Evaluating the quality of automatic, semi-automatic or manually constructed ontologies is of high importance and high challenging. Two different ontologies were evaluated showing encouraging results. In particular, a method for the automatic construction of lexical-syntactic patterns was presented with the aim of discovering candidate concepts and relationships which may be further used for validating concepts and relationships of the ontology to be evaluated. Still, a number of experiments to be carried out, however, we considering important to present the current results obtained in the framework of automatic evaluation of ontologies.

References

1. Hebeler, J., Fisher, M., Blace, R., Perez-Lopez, A., Dean, M.: Semantic Web Programming. Wiley (2011)
2. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL-93-04, Knowledge Systems Laboratory, USA (1993)

3. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data driven ontology evaluation. In: Proceedings of International Conference on Language Resources and Evaluation. (2004)
4. Staab, S., Studer, R., eds.: Ontology Evaluation. International Handbooks on Information Systems. Springer (2004) Chapter 13: Gómez- Pérez, Asunción.
5. Cantador, I., Ferández, M., Castells, P.: A collaborative recommendation framework for ontology evaluation and reuse. In: Actas de International Workshop on Recommender Systems, en la 17th European Conference on Artificial Intelligence (ECAI 2006), Riva del Garda, Italia. (2006) 67–71
6. Sleeman, D., Reul, Q.: CleanONTO: Evaluating Taxonomic Relationships in Ontologies. In Vrandecic, D., Mari, Gangemi, A., Sure, Y., eds.: Proceedings of 4th International EON Workshop on Evaluation of Ontologies for the Web, Edinburgh, Scotland (2006)
7. García-Ramos, S., Otero, A., Fernández-López, M.: Ontologytest: A tool to evaluate ontologies through tests defined by the user. In: Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. IWANN '09, Berlin, Heidelberg, Springer-Verlag (2009) 91–98
8. Völker, J., Vrandečić, D., Sure, Y., Hotho, A.: Aeon - an approach to the automatic evaluation of ontologies. Applied Ontology **3** (January 2008) 41–62
9. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Proceedings of European Knoeledge Ackquisition Workshop (EKAW). (2002)
10. Spyns, P., Reinberger, M.L.: Lexically evaluating ontology triples generated automatically from texts. In Gómez-Pérez, A., Euzenat, J., eds.: ESWC. Volume 3532 of Lecture Notes in Computer Science., Springer (2005) 563–577
11. Brank, J., Mladenić, D., Grobelnik, M.: Gold standard based ontology evaluation using instance assignment. In: Proceedings of the 4th Workshop on Evaluating Ontologies for the Web (EON2006). (2006)
12. Netzer, Y., Gabay, D., Adler, M., Goldberg, Y., Elhadad, M.: Ontology Evaluation through Text Classification. Springer-Verlag, Berlin, Heidelberg (2009)
13. Murdock, J., Buckner, C., Allen, C.: Two methods for evaluating dynamic ontologies. In: Proceedings of the 2nd International Conference on Knowledge Engineering and Ontology Development (KEOD) Valencia, Spain. (2010)
14. Yao, L., Divoli, A., Mayzus, I., James, E.A., Rzhetsky, A.: Benchmarking ontologies: Bigger or better? PLoS Computational Biology **7**(1) (01 2011) 1–15
15. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Ontology evaluation and validation - an integrated formal model for the quality diagnostic task. Technical report, LOA , ISTC-CNR (2005)
16. Salem, S., AbdelRahman, S.: A multiple-domain ontology builder. In: Proceedings of the 23rd International Conference on Computational Linguistics. COLING '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 967–975
17. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. In: Proceedings of the 3rd European Semantic Web Conference (ESWC2006), number 4011 in LNCS, Budva, Springer (2006)
18. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK (1994)
19. Choi, S.S., Cha, S.H., Tappert, C.: A Survey of Binary Similarity and Distance Measures. Journal on Systemics, Cybernetics and Informatics **8**(1) (2010) 43–48