SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

ISSN: 1870-4069

Vol. 119

Technological Trends in Computing

Mario Aldape Pérez
Itzamá López Yáñez
Miguel G. Villarreal Cervantes

Technological Trends in Computing

Research in Computing Science

Series Editorial Board

RESEARCH IN COMPUTING SCIENCE, *Año 15, Volumen 119, 07 de Septiembre de 2015, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https:// www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2005-121611550100-102. ISSN: en trámite, otorgado por el Instituto Nacional del Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 07 de Septiembre de 2015.*

RESEARCH IN COMPUTING SCIENCE, *Year 15, Volume 119, September 07, 2015, is a monthly publication edited by the National Polytechnic Institute through the Center for Computing Research. Av. Juan de Dios Bátiz S/N, Esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, C.P. 07738, Mexico City, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor in charge: Dr. Grigori Sidorov. Reservation of Exclusive Use Rights of Title No. 04-2005-121611550100-102. ISSN: pending, granted by the National Copyright Institute. Responsible for the latest update of this issue: the Computer Research Center, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Last modified on September 07, 2015.*

Technological Trends in Computing

Mario Aldape Pérez
Itzamá López Yáñez
Miguel Gabriel Villarreal Cervantes (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2016

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, Ciudad de México, México

http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

The appeareance, development and success of computational tools and processes have brought about innovation in manifold topics of science and technology worldwide. Science and technology advances to fulfil mankind needs, have lately been triggered by innovation and development in computing science. The accelerated development pace of computing research has impinged on the increasing variety of topics on computational science.

This special issue of *Research in Computing Science* is titled Technological Trends in Computing. It contains 7 articles that were selected and based on a rigorous review process conducted by members of our Editorial Board. It is noteworthy that the content of this issue is a comprehensive set of results related to research projects and technological development. In this volume we present scientific contributions on topics of interest to the community involved in computing, engineering and related areas.

October, 2016

Mario Aldape Pérez
Itzamá López Yáñez
Miguel G. Villarreal Cervantes

Table of Contents

Game Strategy in the NPCs using Interpreted Petri Nets

Alejandra Santoyo-Sanchez[1], Luis Isidro Aguirre-Salas[2],
Carlos Alberto De Jesús-Velásquez[3] and M.V. Alvarez-Ureña[4],

[1]Department of Computing, Universidad de Guadalajara – CUCEI, Guadalajara, Jalisco, México
[2]Department of Engineering, Universidad de Guadalajara – CUCSUR, Autlán de Navarro, Jalisco, México
[3]Compatibility Validation, Intel Tecnología de México S.A., Tlaquepaque, Jalisco, México
[4]Department of Industrial Engineering, Universidad de Guadalajara – CUCEI, Guadalajara, Jalisco, México

Phone (33) 33485900 E-mail: alejandra.santoyo@cucei.udg.mx

Abstract. A common limitation of adventure and strategy games is that players quickly learn the positions and behavior of the characters programmed or Non-Players Characters (NPCs). In this work, is explored using the feedback supervisory control for Discrete Event Systems (DES) to control the actions of the NPCs based on predicting where the player is and decide what to do next. That is, to locate and plan their actions for themselves. Thus, the NPC may present a different behavior every time the player plays.

Keywords: Feedback supervisory control, Non-player characters, Petri Net, Artificial Intelligence.

1 Introduction

In the sort of video games of adventures and strategies, most NPCs are limited by a restricted range of reactions programmed by game designers. In particular, when a player meets the same NPC again after a certain period time, the NPC takes same responses or only reactions that have been manually programed a priori. This model has attracted increasing interest in the use of computational intelligence techniques to control the actions of the NPCs instead of relying on simple heuristics or rules-based systems, as Artificial intelligence (AI) [1], finite state machines [2], [3], search algorithms like A * [4], [5], [6], methods based on software engineering [7], [8], dynamic tables of probability [9] among others.

Manual editing of intelligent behaviors for Non Player Characters (NPCs) of games is a cumbersome task that needs experienced designers, our research aims to assist designers in this task. In [10] Interpreted Petri Nets (IPNs) are used to model and to specify all the possible behaviors of NPCs to avoid blocking situations. The model allows capturing behaviors like: causal relationship, synchronizations, asynchronies, exclusions, concurrence or parallelism, among others. In addition, they

have a mathematical support that makes suitable for analysis of qualitative and quantitative properties. While as in this paper, we focus on dynamic retrieval and selection of behaviors taking into account the current state and the underlying goals (reach the user player). The global behavior of the NPC is dynamically built at runtime the feedback supervisory control for Discrete Event Systems (DES). This paper is organized as follows. Section 2 reviews Petri nets (PN) and IPN notation and concepts used in this article. Section 3 presents the contribution of this paper methodology for the NPC design; i.e. the game strategy for NPC's. Next section 4 presents a case of study to illustrate the use of the game of strategy in the design of a video game. Finally, section 5 provides conclusions and future work.

2 Petri nets and Interpreted Petri nets concepts and properties

This section presents a review the main concepts of the PN and IPN formalism used in this paper. An interested reader can consult [11], [12] and [13] for more details.

2.1 Petri nets

Definition 1. A PN system is a pair (N,M_0) where $N=(P,T,I,O)$ is a bipartite digraph that specifies the net structure and $M_0 : P \to Z^+$ is the initial marking. Each element of N is defined as follows $P=\{p_1,p_2,\ldots,p_n\}$ is a finite set of places; $T=\{t_1,t_2,\ldots,t_m\}$ is a finite set of transitions; $I : P\times T \to Z^+$ and $O : P\times T \to Z^+$ are functions representing the weighted arcs going from places to transitions and from transitions to places, respectively. The initial marking of PN M_0 is a function that assigns to each place of N a non-negative number of tokens, depicted as black dots inside the places.

A PN structure N can be represented by its incidence matrix $C=[c_{i,j}]_{n\times m}$, where $c_{i,j}=O(p_i,t_j)-I(p_i,t_j)$. The sets ${}^\bullet t_j = \{ p_i \mid I(p_i,t_j)\neq 0 \}$ and $t_j{}^\bullet = \{ p_i \mid O(p_i,t_j)\neq 0 \}$ are the set of input and output places of a transition t_j respectively, which are denominated predecessors and successors of t_j respectively. Analogously, the sets of input and output transitions of a place p_i are ${}^\bullet p_i = \{ t_j \mid I(p_i,t_j)\neq 0 \}$ and $p_i{}^\bullet = \{ O(p_i,t_j)\neq 0 \}$ respectively. In a PN system, a self-loop is a relation where $c_{i,j}=O(p_i,t_j)-I(p_i,t_j)=0$ and $O(p_i,t_j)\neq 0$, $I(p_i,t_j)\neq 0$. In this work the self-loop structure is represented by the matrix $F=[f_{i,j}]_{r\times m}$, where $f_{i,j}=O(p_i,t_j)\wedge I(p_i,t_j)$, and r is the number of places with self-loops.

A vector often represents the marking at the k-th instant $M_k=[M_k(p_1)\ \ M_k(p_2)\ \ \cdots\ \ M_k(p_n)]^T$. Hereafter, a marking M can be represented by a list $M=[1^{M(p_1)},2^{M(p_2)},\cdots,i^{M(p_i)},\cdots,n^{M(p_n)}]$ where i-th item is omitted if

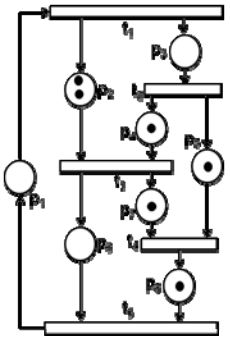
$M(p_i)=0$ and exponents $M(p_i)=1$ are also omitted. For example, a marking $M=[2\ \ 0\ \ 1\ \ 1]^T$ can be represented by the list $M=[1^2,3,4]$.

A transition t_j is enabled at marking M_k if $\forall p_i \in P$, $M_k(p_i) \geq I(p_i,t_j)$; when an enabled transition t_j is fired, then a new marking M_{k+1} is reached. This new marking is computed as $M_{k+1}=M_k+Cv_k$, where v_k is an m-entry firing vector whit $v_k(j)=1$ when t_j is fired once and $v_k(i)=0$ if $i \neq j$ and t_j is not fired, v_k is called Parikh vector; the equation $M_{k+1}=M_k+Cv_k$ is called the PN state equation.

A firing sequence of a PN system (N,M_0) is a transition sequence $\sigma=t_it_j\ldots t_k$ such as $M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \cdots \xrightarrow{t_k} M_k$.. The firing language of (N,M_0) is the set $L(N,M_0) = \{\sigma | \sigma = t_it_j\ldots t_k \wedge M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \cdots \xrightarrow{t_k} M_k$, while the Parikh vector $\bar{\sigma}: T \rightarrow (Z^+)^m$ of σ maps every $t \in T$ to the number of occurrences of t in σ. The fact of reaching M_k from M_0 by firing an enabled sequence σ is denoted by $M_0 \xrightarrow{\sigma} M_k$. The set of all reachable markings from M_0 is $R(N,M_0) = \{M_k | M_0 \xrightarrow{\sigma} M_k$ and $\sigma \in L(N,M_0)\}$ and it is called reachability set.

Definition 2: A p-invariant *Y* of a PN is a rational solution to equation $Y^TC=\bar{0}$. Support p-invariant Y_i is set $|Y_i| = \{p_j | Y_i(p_j) \neq 0\}$.

Example 1: Consider the PN of Fig. 1a. The net consists of 8 places $P=\{p_1,p_2,\ldots,p_8\}$ and 5 transitions $T=\{t_1,t_2,\ldots,t_5\}$. The incidence matrix is illustrated in Fig. 1b. The sets of input and output places of t_1 are $\bullet t_1 = \{p_1\}$ and $t_1 \bullet = \{p_2,p_3\}$ respectively. The initial marking is $M_0=[0\ \ 2\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1]^T$ or $M_0=[2^2,4,5,7,8]$. The set of enabled transitions at M_0 is $E(M_0)=\{t_3,t_4\}$. When transition t_3 fires the net reaches the marking $M_1=[5,6,7^2,8]$. A p-invariant of the system A is $Y_0=[2\ \ 1\ \ 0\ \ 0\ \ 0\ \ 2\ \ 0\ \ 0]$ and its support is $|Y_0| = \{p_1, p_2, p_6\}$.■



$$C=\begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 2 & 0 & -2 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

(a) **(b)**

Fig. 1. a) Petri Net System A, **b)** Incidence matrix of Petri Net System A

2.2 Interpreted Petri nets

Definition 3. An IPN system is a 6-tuple $Q = (N', \Sigma, \Phi, \lambda, \Psi, \varphi)$ where $N' = (N, M_0)$ is a PN system; $\Sigma = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the input alphabet, where α_i is an input symbol; $\Phi = \{\delta_1, \delta_2, \ldots, \delta_s\}$ is the output alphabet of the net, where δ_i is an output symbol; $\lambda : T \to \Sigma \cup \{\varepsilon\}$ is a function that assigns an input symbol to each transition of the net, with the following constraint: $\forall t_j, t_k \in T$, $j \neq k$, if $\forall p_i \quad I(p_i, t_j) = I(p_i, t_k) \neq 0\}$ and both $\lambda(t_j) \neq \varepsilon$, $\lambda(t_k) \neq \varepsilon$, then $\lambda(t_j) \neq \lambda(t_k)$. In this case, ε represents an internal system event. If $\lambda(t_i) \neq \varepsilon$ then transition t_i is said to be controlled, otherwise uncontrolled. T_c and T_u are the sets of controlled and uncontrolled transitions, respectively. $\Psi : P \to \Phi \cup \{\varepsilon\}$ is a labeling function of places that assigns an output symbol or the null event ε to each place of the net as follows: $\Psi(p_i) = \delta_k$ if p_i represents an output signal, in otherwise $\Psi(p_i) = \varepsilon$. In this case $P_m = \{p_i \mid \Psi(p_i) \neq \varepsilon\}$ is the measurable place set and $q = \mid P_m \mid$ is the number of measured places. While $P_{nm} = P \setminus P_m$ is the set of non-measured places.

Finally, $\varphi : R(N, M_0) \to (Z^+)^q$ is a function that associates an output vector to every reachable marking of the net as follows: $\varphi(M_k) = M_k \mid_{Pm}$, where $M_k \mid_{Pm}$ is the projection of M_k over P_m i.e. if $M_k = [M_k(p_1) \quad M_k(p_2) \quad \cdots \quad M_k(p_n)]^T$ and $P_m = \{p_i, p_j, \ldots, p_h\}$ then $M_k \mid_{Pm} = [M_k(p_i) \quad M_k(p_j) \quad \cdots \quad M_k(p_h)]^T$. Notice that function φ is linear and can be represented as a matrix $\varphi = [\varphi_{ij}]_{q \times n}$, where each row $\varphi(k, \bullet)$ of this matrix is an elementary vector where $\varphi(k, i) = 1$ if place p_i is the *k-th* measured place and otherwise $\varphi(k, i) = 0$ and it is called non- measured.

In this paper, a measured place is depicted as an unfilled circle, while a non-measured place is depicted as a filled circle. Similarly, uncontrollable transitions are depicted by filled bars and controllable transitions are depicted by unfilled bars. Also, (Q, M_0) will be used instead of $Q = (N', \Sigma, \Phi, \lambda, \Psi, \varphi)$ to emphasize the fact that there is an initial marking in an IPN.

Example 2: Consider the IPN shown in Fig. 2. The input and output alphabets are $\Sigma = \{a, b\}$ and $\Phi = \{\delta_1, \delta_2, \delta_3\}$ respectively. Functions Ψ and λ are given by:

i	1	2	3	4	5	6	7	8
$\Psi(p_i)$	δ_1	ε	ε	ε	δ_2	δ_3	ε	ε

(**1**)

k	1	2	3	4	5
$\lambda(t_k)$	a	ε	ε	ε	b

(**2**)

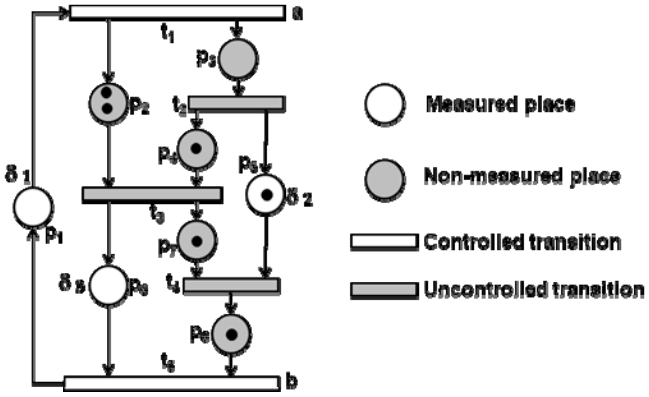


Fig. 2. Interpreted Petri Net System A.

Thus, the controlled transitions are $T_c = \{t_1, t_5\}$ and the uncontrolled ones are $T_u = \{t_2, t_3, t_4\}$. The measured places are $P_m = \{p_1, p_5, p_6\}$ and the non-measured are $P_{nm} = \{p_2, p_3, p_4, p_7, p_8\}$. In this case, the output function is the matrix:

$$\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3)$$

The initial output is $y_0 = \varphi(M_0) = [0 \quad 1 \quad 0]^T$. ■

Similarly to a PN, in an IPN system, a transition t_j is enabled at making M_k if $\forall p_i \in P$; $M_k(p_i) \geq I(p_i, t_j)$ however t_j has fire conditions. When t_j is enabled and t_j is controllable for that t_j fire, it is necessary that the input signal $\lambda(t_j) \neq \varepsilon$ must be given as input. Otherwise when enabled transition t_j is uncontrollable, then it can be fired. In both cases, when a transition t_j is fired a new marking $M_{k+1} = M_k + C(\bullet, t_j)$ is reached and the output symbol $y_{k+1} = \varphi(M_{k+1})$ is observed. Also, the firing sequence and firing language is computed and denoted as in PN system to enhance the fact that there exists an initial marking in an IPN $Q = (N', \Sigma, \Phi, \lambda, \Psi, \varphi)$ in this paper it is denoted as (Q, M_0).

3 Game Strategy Description

In this paper, Step Sate-Feedback Supervisory Control [14] and Error Petri Net are used for compute game strategy for each NPC. Based on these mechanisms and

according to Supervisory Control Theory from [15], our Step Sate-Feedback Supervisory Control consists of three elements (Fig. 3):

1) A Dynamic Discrete event system (DES), representing the system to be controlled; in this work the system itself is composed by the opponents.
2) The required behavior for the system, called specification; in this case all behavior possible of the player (the user game).
3) An external agent, called supervisor, which restricts the behavior of the system to the behavior of the specification by the manipulation of controllable events.

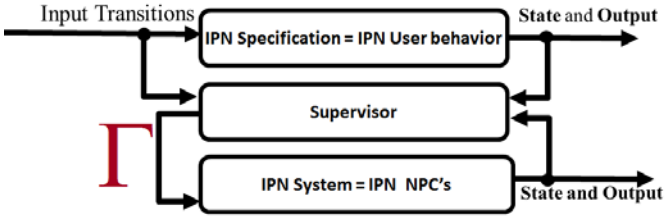


Fig. 3. Game Strategy Scheme.

In this work, the model system (NPC's) and specification (user game) are IPN models, where it's output information indicates which output information the system must reach. Then the enabling rule of events Γ determines which subset of inputs events are enabled to occur into the system model (to generate the behavior of these adversaries), based in the input string generated by the user game, and the state and outputs of both system and specification model.

A novel technique for model NPC's based on IPN and controlled place is presented in [10]. The procedure includes a method to capture restriction operation in term of the linear equations. The IPN model obtained with that methodology is used in this approach to establish the control law based Step Sate-Feedback Supervisory Control to simulate intelligence in the NPC's as follow.

3.1 Modeling the specification: user behavior

In this paper all behavior possible of the player (the user game) and opponents (NPC's) are modeled using the propose methodology in [10]. That proposal based on IPN and controlled place, and the model obtained is composed by all the IPN model of the elements: player and opponents, also the function φ is the identity matrix, due to the game is composed only by measured places. In the strategy games the player and opponents do not always have a complete knowledge of the system state; in these cases, the function φ represents only the places that each player and opponent can measure. In this paper all behavior possible of the player (the user game) is modeled using the follow algorithm.

Algorithm 1: Modeling the specification: user behavior.

Inputs: A model in IPN computed with the algorithm [10].
Outputs: Specification (IPN model of the player).
Procedure:

1. **Identify the player of the game and generate "system specification".** In this step the player is defined as the system specification, this is done by selecting the places and transitions of the net which correspond to the player, each NPC will try to achieve the same token marking as the specification.
2. **Defining the knowledge places for the player.** Each one of these places is selected according to the game specification. It's defined a matrix φ_i. Consider for example the state variable *position* for the player, in this case the relevant values are: $room_1$, $room_2$, …, $room_7$ where there can be non-measurable rooms for the player. Thus the IPN of player is defined as: (Q, M_0^{player})

3.2 Modeling the system: NPC behavior

In a similar form, to generate the behavior of these adversaries, the IPN of the close loop system with the supervisor proposed in [10] is analyzed.

Algorithm 2: Modeling the system.

Inputs: A model in IPN of the close loop system with the supervisor [10].
Outputs: The system to be controlled (IPN model of the opponent).
Procedure:

1. **Composition of the system.** In this step the system is defined, this is done by selecting the places and transitions of the net that do not correspond to the player.
2. **Defining the knowledge places for each NPC.** Each one of these places is selected according to the game specification. It´s defined a matrix φ for each NPC. Consider for example the state variable *position* for an opponent, in this case the relevant values are: $room_1$, $room_2$, …, $room_7$ where there can be non-measurable rooms for this NPC only. In this case the IPN for each NPC is defined as (Q, M_0^{NPCi}).

The next algorithm uses the individual marking of each part of the system and specification to define the strategy game.

Algorithm 3: Measuring the system and specification.

Inputs: Specification (IPN model of the player) and a system to be controlled (IPN model of the opponent)
Outputs: A vector containing the observable information and a transition-firing vector.
Procedure:

1. **Selecting the NPC.** It's used the φ_i of the NPC whose information is required and the system marking that are necessary for next steps.
2. **Obtaining the output function of the system.** Define the output for φ_i for the system and the output for the specification. By applying the equation: $\varphi_i(M_j^{NPCi})$.
3. **Obtain the output function of the specification**. Multiply the φ_i and the player marking M_k^{player} for the system and the output for the specification. By applying the equation: $\varphi_i(M_k^{player})$
4. **Obtaining the vector of information.** Subtract the system output from the specification output. By applying the equation: $\varphi_i(M_k^{player}) - \varphi_i(M_j^{NPCi}) = M_{sig}$.
5. **Calculating the firing vector.** Select the route for the NPC to follow by calculating the firing vector that allows the NPC to reach the desired destination. Let $M_k \rightarrow M_{(k+1)}$ executing the transition t_i be the marking of the system and $M_l \rightarrow M_{(l+1)}$ executing the firing vector w_i be the specification marking where $w_i = \{t_a, t_b, t_c, \ldots, t_s\}$. Then $M_k \rightarrow M_{k'} by\, t_a, M_{k'} \rightarrow M_{k''} by\, t_b, \ldots, \rightarrow M_{(k+1)} by\, t_s$. Calculate the firing vector w_i for the system solving the next equation system: $CV = M_{sig}$, where C is the incidence matrix of the NPC_i and $V = [v_1, v_2, \ldots, v_n]$ where V_i is the amount of firing times necessary of t_i to reach the desired state if the player remains static, in other case is necessary to apply this algorithm again until the player is captured by an opponent or the player meets its goal objectives.

If M_{sig} is a vector 0, means that the player and the NPC_i both are in the same room or both are in not observable rooms, if $\exists \alpha_j \in M_{sig} \mid a_j > 0$ then the player is visible for the NPC_i and in any other case the player is not visible for the NPC_i.

The application from all algorithms shown in this section is illustrated in the next case study.

4 Case of study strategy game

Fig. 4 represents the strategy game considered in this case of study proposed in [10]. It is a dungeon environment, which is composed by seven rooms (h_1, h_2, …, h_7). Inside a dungeon there are the following elements: a player (j_1), two opponents (o_A and o_B), and a reward. The behavior of each element inside dungeon is the following. Player (j_1) has the goal of finding the reward and exit from the dungeon. Also, player should achieve its goal just by avoid being in the same room as any of opponents. This strategy game is indicated by the user game. Two opponents (o_A and o_B) have the goal of finding the player. In this case its strategy game consists in moving and watching between dungeon's rooms, which is computed by an algorithm.

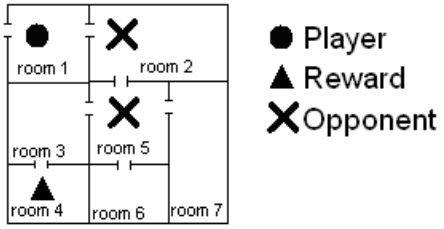


Fig. 4. Component of the game [10].

Fig. 5 show IPN model game computed using the algorithm proposed in [10]. Where the input and output alphabets are: $\Sigma = \{a_{1-2}, a_{2-1}, a_{2-5}, a_{3-4}, a_{3-5}, a_{4-3}, a_{5-2}, a_{5-3}, a_{5-6}, a_{5-7}, a_{6-5}, a_{7-5}\}$ and $\Phi = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7\}$ respectively. Functions Ψ and λ are given by:

i	1	2	3	4	5	6	7
$\Psi(p_i)$	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7

(**4**)

k	*1, 13, 25*	*2, 14, 26*	*3, 15, 27*	*4, 16, 28*	*5, 17, 29*	*6, 18, 30*	*7, 19, 31*	*8, 20, 32*	*9, 21, 33*	*10, 22, 34*	*11, 23, 35*	*12, 24, 36*
$\lambda(t_k)$	a_{1-2}	a_{2-1}	a_{2-5}	a_{5-}	a_{3-5}	a_{5-3}	a_{4-3}	a_{3-4}	a_{5-6}	a_{6-5}	a_{7-5}	a_{5-7}

(**5**)

All transitions are controlled and all places are measured. Thus the output function φ is an identity matrix. In the final step it is necessary to synchronize the state variables, in this case it step is not necessary.

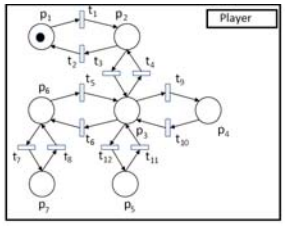


Fig. 5.a) Player IPN [10]

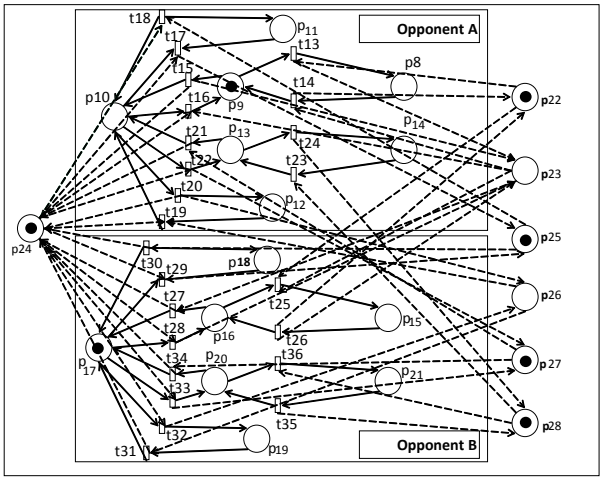


Fig. 5.b) IPN Control places among opponet A and opponent B [10].

Fig. 5. IPN model computed with [10].

While as, the Fig. 5a and Fig. 5b show IPN model of player and the two opponents, which are obtained using the algorithms 1 y 2 respectively. In this case, the output function for the player and opponents are the matrix:

$$\varphi_{NPCA} = \begin{bmatrix} 1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \quad (6)$$

$$\varphi_{NPCB} = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \end{bmatrix} \quad (7)$$

The marking are:

$$M_0^{player} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

$$M_0^{NPCA} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T \quad (8)$$

$$M_0^{NPCB} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

Now the algorithm 3 is used, in this case for the opponent A $\varphi_{NPCA}(M_0^{player}) = [1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$ and $\varphi_{NPCA}(M_0^{NPCA}) = [0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$; then

$\varphi_{NPCA}(M_0^{player}) - \varphi_{NPCA}(M_0^{NPCA}) = M_{sig} = [1 \quad -1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$. The firing vector computed is: $\sigma = t_{14}$. In similar form is computed by the opponent B. Then the enabling rule of events Γ determines $\lambda(t_{14}) = a_{2-1}$, thus is defined the behavior for each NPC´s.

5 Conclusions

This paper presents a novel technique for dynamic retrieval and selection of behaviors taking into account the current state and the underlying goals (reach the user player). The global behavior of the NPC is dynamically built at runtime the feedback supervisory control for Discrete Event Systems (DES). In proposal, the use of Interpreted Petri Nets (IPN) which is an extension of Petri Nets (PN) for design the Intelligence in the NPCs is explored. The proposal is based creating a relationship between input signals and output signals for IPN models of each NPCs to define its behavior.

As future work this methodology will analyze to establish the control law based in Feedback Supervisory Control and Partial Observation to simulate intelligence in the NPC inside a collaboration schema.

6 References

[1] M. van Lent, "Game Smarts", *Computer*, Vol. 40, pp 99-101, April 2007.

[2] W. M. Wonhan, and P.J. Ramadge, "On the Supremal Controllable Sublanguage of a Given Language", *SIAM Journal on Control and Optimization*, Vol. 25, No.3, pp. 635-659, 1987.

[3] J.E. Hopcroft and J.D. Ullman. "Introduction to automata theory, languages, and computation", Ed. Addison-Wesley, 1979 [4] P.E. Hart, L.J. Nilsson and B. Raphael. "A formal basis for the heuristic determination of minimum cost paths", *IEEE Transactions on Systems Science and Cybernetic*s, Vol. 4, No. 2, pp. 100-107, 1968.

[5] P.E. Hart, L.J. Nilsson, and B. Raphael. Correction to "A formal basis for the heuristic determination of minimum cost paths", *SIGART Newsletter*, No. 37, pp. 28-29, 1932.

[6] V. Kumar, and D. S. Nau. "A general branch-and-bound formulation for AND/OR graph and game tree search", *Search in Artificial Intelligence*, pp. 91-130, Ed. Springer-Verlag, 1988.

[7] M. McNaughton, M. Cutumisu, D. Szafron, J. Schaeffer, J.Redford, and D. Parker. "ScriptEase: Generating Scripting Code for Computer Role-Playing Games", *in Proc.19th IEEE International Conference on Automated Software Engineering ASE´04*, 2004, Linz, Austria, pp. 386 – 387.

[8] L. Wei-Po, L. Li-Jen, and C. Jeng-An; "A Component-Based Framework to Rapidly Prototype Online Chess Games for Home Entertainment"; *in Proc. IEEE International Conference on Systems, Man, and Cybernetics* SMC'06, 2006, Taipei, Taiwan, Vol. 5, pp 4011-4016, 2006.

[9] W. Yingxu. "Mathematical models and properties of games". *in Proc. Of the Fourth IIEEE Conference Cognitive Informatics ICCI´05*, 2005, Irvine, CA, USA, pp. 294-300.

[10] Santoyo-Sanchez, A.; Pérez-Martinez, M.A.; De Jesús-Velásquez, C.; Aguirre-Salas, L.I.; Alvarez-Ureña, M.A.; "Modeling methodology for NPC's using interpreted Petri Nets and feedback control", *in Proc. 7th International Conference on IEEE Electrical Engineering Computing Science and Automatic Control (CCE)*, 2010, Tuxtla Gutiérrez, Chiapas, México, pp.369 – 374, 2010.

[11] T. Murata, "Petri nets: Properties, analysis, and application", *in Proc. Of the IEEE*, Vol. 77, No.4, pp. 541-580, 1989.

[12] J. Desel, J. Esparza and C. J. van Rijsbergen, *Free choice Petri nets*, pp. 1-5, Ed. Cambridge University Press, 2005.

[13] M. E. Meda, A. Ramirez and A. Malo, "Identification in discrete event systems", *in Proc.1998 IEEE International Conference Systems, Man and Cybernetics*, *SMC 1998*,*San Diego CA*., pp. 740-745.

[14] A. Santoyo-Sanchez, A. Ramírez-Treviño, C. De Jesús Velásquez, L.I. Aguirre-Salas, "Step State-feedback Supervisory Control of Discrete Event Systems using Interpreted Petri Nets", *in Proc.13th IEEE International Conference on Emerging Technologies and Factory Automation*, *ETFA 2008*, Hamburg Germany, pp. 926 – 933.

[15] P.J. Ramadge, and W. M. Wonhan, "Supervisory Control of a Class of Discrete Event Processes", *SIAM Journal on Control and Optimization*, Vol. 25, No.1, pp. 206-230, 1987.

[16] De-Jesus, C.A.; Ramirez-Trevino, A.; "Controller and observer synthesis in discrete event systems using stability concepts", *in Proc. IEEE International Conference on Systems, Man, and Cybernetics*, 2001, Tucson, AZ, USA, Vol. 1, pp. 664 – 668.

An experimental comparison of supervised classification algorithms for breast cancer detection

David González-Patiño[1], Yenny Villuendas-Rey[2], Amadeo J. Argüelles-Cruz[1]

[1]Centro de Investigación en Computación del Instituto Politécnico Nacional, Avenida Juan de Dios Bátiz esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, Gustavo A. Madero, CP 07738, Ciudad de México, D.F., México
davidglezp-92@hotmail.com; aarguelles@ipn.mx

[2]Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n, Nueva Industrial Vallejo, Gustavo A. Madero, 07700 Ciudad de México, D.F., México
yenny.villuendas@gmail.com

Abstract. Breast cancer is considered as one of the most common cancers worldwide. It affects millions of women, of all ages. The automatic or semi-automatic detection of breast cancer lesions is still a challenge for the scientific community, and several methods have been proposed to perform this task. In this paper, we evaluate the performance of several supervised classification algorithms, for breast cancer detection. We also explore the impact of rough Set based feature selection technique in breast cancer data obtained from mammography images. The experimental results show that bio-inspired techniques based on Artificial Immune Systems obtain promising results.

Keywords: breast cancer · supervised classification · experimental comparison.

1 Introduction

The most common cancers among the world are breast, colon, lung and prostate cancer [1]. Breast cancer is a serious problem in women's health around the entire world. According to the paper presented by Felicia Marie Knaul and collaborators [2], since 2006, breast cancer is the worldwide second cause of death among women between 30 and 54 years. In Mexico, an increase of 16,500 annual cases is estimated by 2020, due to population growth and aging. Breast cancer causes many more deaths compared to cervical cancer [2], and affects adult women of all ages and all economic levels. Aging is a factor in the development of cancer; however, age is not a main factor given that breast cancer does not depend directly on it but certain types of cancer occur more frequently at older ages. Therefore, it is highly recommended to perform an annual examination for breast lesions for women aged between 40 and 49.

There is a necessity in all countries to promote the use of techniques to detect breast cancer in early phases. The data presented [1] shows that only between 5% and 10% cases of breast cancer in Mexico are detected in early phases compared to 50% of the cases detected in the United States of America. The automatic or semi-

automatic detection of breast cancer lesions is still a challenge for the scientific community [3-5]. Several works have been proposed for breast image segmentation and classification [6-8], and international repositories of breast cancer data are now available [9-11]. However, there is a lack of recent comparative studies in the performance of supervised classification algorithms for detecting breast cancer. In this paper we address this issue, by performing an experimental comparison of several classification algorithms over mammographic image data. Our comparison includes classic algorithms and bio-inspired algorithms, which presented competitive performance compared to classic algorithms. We also explore the influence of feature selection methods in breast cancer classification.

The paper is organized as follows. Section 2 addresses some previous works in the field of automatic or semi-automatic breast cancer diagnosis. Section 3 explains the classification algorithms under comparison, while section 4 presents the experimental results. The paper finalizes with conclusions and future work.

2 Previous Works

Breast cancer is a serious problem in women's health around the entire world. There are several techniques used for the accurate diagnosis of breast cancer, such as mammography studies [12], Magnetic Resonance Image (MRI) [13], ultrasounds [14] and biopsy [15], among others. The automatic and semi-automatic detection of breast cancer lesions have been addressed since 1987 [16]. Later in 1993 [17] a method was proposed for cancer diagnosis based on image analysis. In 1999, Pena-Reyes and Sipper proposed an automatic diagnosis of breast cancer combining two methodologies: fuzzy systems and evolutionary algorithms [18]. In 2015 a work was presented [19] using Krill Herd optimization algorithm to classify breast cancer datasets. This work obtained a simple classification rule in order to classify breast cancer lesions. This rule can be used in the decision making process for the breast cancer diagnosis. In 2016 Magna et al. proposed an ensemble of classification models based on artificial immune systems to identify mammography anomalies for breast cancer detection [20]. This resulted in promising results using it for classification tasks, which represents an important advance in artificial immune systems.

Nevertheless, there is no a comparison of algorithms used for classification task including bio-inspired algorithms. In this paper, we addressed this issue, by performing a wide-range comparison of several supervised classifications algorithms, for detecting the presence or absence of breast-cancer in mammography images data.

3 Classification algorithms used in the experiments

Supervised classifiers aim at predicting the class of a new, unlabeled instance, by considering the information of previously labeled instances; that is, the instances in

the training set. In the following section, we will explain the main aspects of the supervised classification algorithms under comparison.

The Nearest Neighbor (NN) classifier was proposed by Cover and Hart in 1967 [21] as an algorithm to classify patterns according to the k nearest neighbors in the previously learnt patterns. This algorithm is one of the most fundamental and simpler classification methods, positioning it as one of the first choices for a classification task.

It has been used to solve a wide range of problems, such as on-chip template reduction [22], study of unfolded state of peptides and proteins [23], traffic flow forecasting [24] and breast cancer classification [25].

Naïve Bayes classifier is based on the Bayes Theorem and is useful to deal with issues of uncertainty and noise [26].

Naïve Bayes assumes that all the attributes are not related and even with that assumption the Naïve Bayes classifier has showed a nice performance for estimations and feature selection for text classification [27]. This classifier has also been used for risk classification [28], fault diagnosis [29] and recently for breast cancer detection [30].

Support Vector Machines (SVM) have been gaining popularity within pattern recognition [31-33]. SVM has been used for recognizing human actions [31], protein classification [32], and also for breast tumor classification [33].

Decision trees are a family of supervised classifiers. A decision tree produces a model with rules for both continuous and categorical variables [34]. Among decision trees, the C4.5 algorithm is highly relevant [35]. This method is used for predicting categorical outcomes and it has been used for imbalanced data sets and redundant features in text categorization [36]. It has also been used for breast cancer prediction in ultrasound images [37], heart disease prediction [38] and satellite network fault prediction [39].

RIPPER is a rule-based classifier [40]. This classifier is one of the most popular algorithms because of the set of rules generated using incremental reduced error. This algorithm is based on finding a set of rules that cover all the patterns for each class in the dataset [40].

Among bio-inspired classification techniques, the ones based on the Immune System have acquired a high relevance for several disease classifications, such as chest diseases [41] breast cancer [20] and Parkinson [41].

The Immune system is a complex system which allows processing information using pattern recognition, learning skills and pattern memory. This system is decentralized, which implies that this system is fault tolerant. The Artificial Immune System tries to mimic the behavior of the Biological Immune System [42].

The Biological Immune System is composed of cells, molecules and organs whose function is to keep a healthy organism, protecting it from harmful agents (Antigens).

The innate Immune System has, since we were born, the ability to recognize and destroy antigens. This system is the responsible for activating the adaptive immune system response.

There are many algorithms based on Artificial Immune Systems which try to mimic some process or characteristic of the Biological Immune System [42]. The most common are:

Immune network: It is based on creating a network that has connected nodes that recognizes; also it has nodes that interact with external antigens [43].

Negative Selection: It is based on discriminating cells which belong and not belong to the system. Also it detects changes in the system behavior using detectors that do not necessarily have communication with other detectors [44].

Clonal Selection: It is based on the idea that only the best lymphocytes responding to the presence of antigens will be reproduced using cloning algorithms [45].

Dendrite cells: It is based on a network that represents the behavior of cells in a population. It is used in multi-scale approach [46].

In this paper we explore two immune-based algorithms and their variations: the Clonal Selection based algorithms [45], and the Artificial Immune Recognition System (ARIS) algorithms [47].

Clonal Selection algorithm is very similar to genetic algorithms but the population is dynamic. Also, only the best antibodies of the population are taken using an affinity function. These antibodies are cloned and mutated using an inverse affinity function (The higher the affinity, the lower the probability of mutation). Antigens are chosen with the highest affinity and the process is repeated.

On the other hand, ARIS algorithm was proposed by Watkins [48] in 2001 as an unsupervised learning algorithm inspired by the biological theory of clonal selection proposed by Burnet in 1957 [49]. The AIRS algorithm was designed later for classification problems [47]. Typically the Euclidean distance is used as affinity function. For this algorithm, all the cells are initialized with small random quantities. All the algorithms were tested in the WEKA software [50] using the implementations of Jason Brownlee in 2015.

4 Experiments

The dataset used in this paper was from the Breast Cancer Digital Repository [10] by the Faculty of medicine in the University of Porto, in Portugal.

This dataset is composed of 200 biopsy-tested lesions of 190 women, rendering 362 segmentations including clinical data and descriptors based on the segmented images.

This is a binary class dataset (Benign class and malignant class) due to the classification made by the radiologist. The dataset has missing values wich are represented as NaN (Not a number). The lesion outlines were identified by a group of expert radiologists.

Using the AIRS classifiers implementation for WEKA software [50], we obtained the performances for the algorithms implemented as showed in Table 3. There are three fundamental models implemented in AIRS classifiers: Based on AIRS algorithm, Based on Clonal Selection algorithm and Based on Immune Systems.

According to Table 3, only the algorithms that had the best performances were used for later tests. The validation algorithm used for all algorithms was 10-fold cross-validation [51].

Table 3. Performances of the bio-inspired algorithms in WEKA using default configurations.

Algorithm	Performance
AIRS 1	70.442%
AIRS 2	67.6796%
AIRS 2 Parallel	71.2707%
CLONALG	56.3536%
CSCA	56.0773%
Immunos 1	56.0773%
Immunos 2	59.3923%
Immunos 99	55.8011%

The algorithms used in the comparisons were AIRS 1 [48], CLONALG [52] and Immunos2 [53] because they showed to be the algorithms with the best performances for each fundamental model. The default parameters and their descriptions of AIRS 1 and CLONALG are presented in Table 4 and Table 5. Immunos2 does not have modifiable parameters.

The parameters in bold letters of Table 4 and Table 5 are very sensitive to modifications in order to obtain a better performance. Each parameter was changed by the values of Table 6. When changing the previous parameters, we obtained the performances shown in Table 7.

The classical supervised classifiers were used with their default parameters. Gathering all the information, we can definitely analyze the performances of the classifications made by all the algorithms tested in this paper. The results are shown in Table 8.

We can observe that AIRS algorithm is the second best algorithm according to the performances shown.

Feature selection has been used for classification tasks as presented in 2016 [54]. However, feature selection was performed using the LEX algorithm [55] for the dataset presented in this paper, although it shown that for this dataset, the performance was lower in all proven cases.

Some of the tests for the reducts [56] are shown in Table 9, the first column shows the index attribute used in the Reduct, and the following columns represent the performances of each bio-inspired algorithm.

As we can deduct from Table 9, using a reduct for the dataset does not result in a better performance. All the results showed that the use of reducts decrease the performance in both bio-inspired algorithms.

Table 4. Parameter descriptions for AIRS algorithm.

Parameter	Description	Default value
Affinity threshold scalar	Used to determine whether or not a candidate memory cell can replace the previous best matching memory cell	0.2
Initial ARB cell pool size	Number of randomly selected training data instances used to seed the ARB cell pool	1
Clonal rate	Determine the number of mutated clones	10
Hypermutation rate	Determine the number of clones a memory cell can create using the clonal rate	2
k-Nearest Neighbor	**The number of best matching memory cells used during the classification stage to majority vote the classification of unknown data patterns**	**3**
Initial memory cell pool size	Specifies the number of randomly selected training data instances used to seed the memory cell pool	1
Mutation rate cloned ARBs	Determine the degree of mutation of cloned ARB	0.1
Total training instances to calculate affinity threshold	Specifies the number of training data instances used to calculate the affinity threshold (mean affinity between data instances)	-1
Stimulation threshold	**Determine when to stop refining the pool of ARBs for an antigen**	**0.9**
Total allocable resources	Specifies the maximum number of resources (B-cells) that can be allocated to ARBs in the ARB pool	150

Table 5. Parameter descriptions for CLONALG algorithm.

Parameter	Description	Default value
Antibody pool size	**Antibodies maintained in the memory pool and remainder pool**	**30**
Clonal factor	**Used to scale the number of clones created by the selected best antibodies**	**0.1**
Total generations	**Total number of times that all antigens are exposed to the system**	**10**
Remainder pool percentage	Percentage of the total antibody pool size allocated for the remainder pool	0.1
Selection pool size	Total number of best antibodies selected for cloning and mutation each iteration	20
Total replacements	**The total number of antibodies in the remainder pool that are replaced each iteration**	**0**

Table 6. Modified parameters descriptions for immune-based algorithms.

AIRS	CLONALG
k-Nearest Neighbor = 1	Antibody pool size = 28
Stimulation threshold = 0.99	Antibody pool size = 28
	Total generations = 15
	Total replacements = 3

Table 7. Performances of the different AIRS algorithms in WEKA using modified configurations in the parameters.

Algorithm	Performance
AIRS	77.9006 %
CLONALG	59.3923 %

Table 8. Performances of all the classification algorithms tested in WEKA ordered by descendant performance.

Algorithm	Performance
SVM (SMO)	80.1105 %
AIRS	**77.9006 %**
JRip	75.1381%
J48 (C4.5)	75.1381 %
3NN	74.8619 %
1NN	73.4807 %
Naïve Bayes	72.3757 %
Immunos	**59.3923%**
CLONALG	**59.3923 %**

Table 9. Performances of the different reducts found by LEX algorithm using modified configurations in the parameters for the bio-inspired algorithms in WEKA.

Reduct	Performance AIRS	Performance CLONALG
{20;2;3;4;8;9;13;15;18;19;0;21;22;23;28;32}	69.6133%	48.3425%
{20;2;3;4;8;9;10;11;12;13;15;18;19;0;21;22;23}	69.6133%	54.6961%
{20;2;3;4;8;9;10;11;13;15;18;19;0;21;22;23;24;27}	71.8232%	53.8674%
{20;2;3;4;8;9;10;11;13;15;18;19;0;21;22;23;25}	70.7182%	53.0387%

5 Conclusions

According to the tests performed in this paper, the bio-inspired algorithms for pre-diagnosis of breast cancer are as competitive as classic algorithms. The use of this kind of algorithms is very useful in the dataset presented as the performances of the bio-inspired algorithms were nearly at the top, just after Support Vector Machines. Adjusting the parameters of the bio-inspired algorithms can result in having a better performance according to the tests made in this paper. The use of Rough Set based feature selection did not show any improvement in classifier performance.

Acknowledgments

The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Investigación en Computación, and Centro de Innovación y Desarrollo Tecnológico en Cómputo), the Consejo Nacional de Ciencia y Tecnología (Conacyt), and Sistema Nacional de Investigadores for their economic support to develop this work

6 References

1. Borja-Aburto, V. H., Dávila-Torres, J., Rascón-Pacheco, R. A., González-León, M., Fernández-Gárate, J. E., Mejía-Rodríguez, I., ...& Escudero-de los Ríos, P. M. (2016). Cancer mortality in the Mexican Social Security Institute, 1989-2013. salud pública de méxico, 58(2), 153-161. (2016)
2. Knaul, F.M., Nigenda, G., Lozano, R., Arreola-Ornelas, H., Langer, A., Frenk, J.: Breast cancer in Mexico: an urgent priority. Salud publica de Mexico 51, s335-s344 (2009)
3. Cecchini, R.S., Swain, S.M., Costantino, J.P., Rastogi, P., Jeong, J.-H., Anderson, S.J., Tang, G., Geyer, C.E., Lembersky, B.C., Romond, E.H.: Body mass index at diagnosis and breast cancer survival prognosis in clinical trial populations from NRG Oncology/NSABP B-30, B-31, B-34, and B-38. Cancer Epidemiology Biomarkers & Prevention 25, 51-59 (2016)

4. Gubern-Mérida, A., Vreemann, S., Martí, R., Melendez, J., Lardenoije, S., Mann, R.M., Karssemeijer, N., Platel, B.: Automated detection of breast cancer in false-negative screening MRI studies from women at increased risk. European journal of radiology 85, 472- 479 (2016)
5. Krawczyk, B., Galar, M., Jeleń, Ł., & Herrera, F. (2016). Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. Applied Soft Computing, 38, 714-726. (2016)
6. Kowal, M.: Computer-aided diagnosis for breast tumor classification using microscopic images of fine needle biopsy. Intelligent Systems in Technical and Medical Diagnostics, pp. 213-224. Springer (2014)
7. Su, H., Shen, Y., Xing, F., Qi, X., Hirshfield, K.M., Yang, L., Foran, D.J.: Robust automatic breast cancer staging using a combination of functional genomics and image-omics. In: Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE, pp. 7226-7229. IEEE, (2015)
8. Gu, P., Lee, W.-M., Roubidoux, M.A., Yuan, J., Wang, X., Carson, P.L.: Automated 3D ultrasound image segmentation to aid breast cancer image interpretation. Ultrasonics 65, 51-58 (2016)
9. Lichman, M.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2013)
10. Moura, D.C., López, M.A.G.: An evaluation of image descriptors combined with clinical data for breast cancer diagnosis. International journal of computer assisted radiology and surgery 8, 561-574 (2013)
11. Van't Veer, L.J., Dai, H., Van De Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T.: Gene expression profiling predicts clinical outcome of breast cancer. nature 415, 530-536 (2002)
12. Wu, Y., Giger, M.L., Doi, K., Vyborny, C.J., Schmidt, R.A., Metz, C.E.: Artificial neural networks in mammography: application to decision making in the diagnosis of breast cancer. Radiology 187, 81-87 (1993)
13. Kriege, M., Brekelmans, C.T., Boetes, C., Besnard, P.E., Zonderland, H.M., Obdeijn, I.M., Manoliu, R.A., Kok, T., Peterse, H., Tilanus-Linthorst, M.M.: Efficacy of MRI and mammography for breast-cancer screening in women with a familial or genetic predisposition. New England Journal of Medicine 351, 427-437 (2004)
14. Warner, E., Plewes, D.B., Hill, K.A., Causer, P.A., Zubovits, J.T., Jong, R.A., Cutrara, M.R., DeBoer, G., Yaffe, M.J., Messner, S.J.: Surveillance of BRCA1 and BRCA2 mutation carriers with magnetic resonance imaging, ultrasound, mammography, and clinical breast examination. Jama 292, 1317-1325 (2004)
15. Kuhl, C.K., Elevelt, A., Leutner, C.C., Gieseke, J., Pakos, E., Schild, H.H.: Interventional breast MR imaging: clinical use of a stereotactic localization and biopsy device. Radiology 204, 667-675 (1997)
16. Wittekind, C., Schulte, E.: Computerized morphometric image analysis of cytologic nuclear parameters in breast cancer. Analytical and quantitative cytology and histology/the International Academy of Cytology [and] American Society of Cytology 9, 480-484 (1987)
17. Wolberg, W.H., Street, W.N., Mangasarian, O.L.: Breast cytology diagnosis via digital image analysis. Analytical and Quantitative Cytology and Histology 15, 396-404 (1993)
18. Pena-Reyes, C.A.s., Sipper, M.: A fuzzy-genetic approach to breast cancer diagnosis. Artificial intelligence in medicine 17, 131-155 (1999)
19. Arumugam, M.M., Kumari, S.: Application of bio-inspired krill herd algorithm for breast cancer classification and diagnosis. Indian Journal of Science and Technology 8, (2015)
20. Magna, G., Casti, P., Jayaraman, S.V., Salmeri, M., Mencattini, A., Martinelli, E., Di Natale, C.: Identification of mammography anomalies for breast cancer detection by an ensemble of classification models based on artificial immune system. Knowledge-Based Systems 101, 60-70 (2016)
21. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Information Theory, IEEE Transactions on 13, 21-27 (1967)

22. Xia, W., Mita, Y., Shibata, T.: A Nearest Neighbor Classifier Employing Critical Boundary Vectors for Efficient On-Chip Template Reduction. (2015)
23. Toal, S.E., Kubatova, N., Richter, C., Linhard, V., Schwalbe, H., Schweitzer-Stenner, R.: Randomizing the Unfolded State of Peptides (and Proteins) by Nearest Neighbor Interactions between Unlike Residues. Chemistry–A European Journal 21, 5173-5192 (2015)
24. Zou, T., He, Y., Zhang, N., Du, R., Gao, X.: Short-Time Traffic Flow Forecasting Based on the K-Nearest Neighbor Model. traffic 1, 36 (2015)
25. Bagui, S.C., Bagui, S., Pal, K., Pal, N.R.: Breast cancer detection using rank nearest neighbor classification rules. Pattern recognition 36, 25-34 (2003)
26. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pp. 338-345. Morgan Kaufmann Publishers Inc., (1995)
27. Chen, J., Huang, H., Tian, S., Qu, Y.: Feature selection for text classification with Naïve Bayes. Expert Systems with Applications 36, 5432-5435 (2009)
28. Minnier, J., Yuan, M., Liu, J.S., Cai, T.: Risk classification with an adaptive naive bayes kernel machine model. Journal of the American Statistical Association 110, 393-404 (2015)
29. Sharma, R.K., Sugumaran, V., Kumar, H., Amarnath, M.: A comparative study of naïve Bayes classifier and Bayes net classifier for fault diagnosis of roller bearing using sound signal. International Journal of Decision Support Systems 1, 115-129 (2015)
30. Kharya, S., Soni, S.: Weighted Naive Bayes Classifier: A Predictive Model for Breast Cancer Detection. International Journal of Computer Applications 133, 32-37 (2016)
31. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, pp. 32-36. IEEE, (2004)
32. Leslie, C.S., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: Pacific symposium on biocomputing, pp. 566-575. (2002)
33. Wu, W.-J., Lin, S.-W., Moon, W.K.: An Artificial Immune System-Based Support Vector Machine Approach for Classifying Ultrasound Breast Tumor Images. Journal of digital imaging 28, 576-585 (2015)
34. Duda, R., Hart, P., Stork, D.: Pattern Classification, Jon Wiley & Sons Inc. New York 630-633 (2001)
35. Quinlan, J.R.: C4. 5: programs for machine learning. Elsevier (2014)
36. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4. 5. In: Proceedings of the twenty-first international conference on Machine learning, pp. 41. ACM, (2004)
37. Kuo, W.-J., Chang, R.-F., Chen, D.-R., Lee, C.C.: Data mining with decision trees for diagnosis of breast tumor in medical ultrasonic images. Breast cancer research and treatment 66, 51-57 (2001)
38. Sharma, P., Saxena, K., Sharma, R.: Heart Disease Prediction System Evaluation Using C4. 5 Rules and Partial Tree. Computational Intelligence in Data Mining—Volume 2, pp. 285-294. Springer (2016)
39. Lin, Y., Ding, S., Wang, Y., Geng, J.: A method of satellite network fault synthetic diagnosis based on C4. 5 algorithm and expert knowledge database. In: Wireless Communications & Signal Processing (WCSP), 2015 International Conference on, pp. 1-5. IEEE, (2015)
40. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning, pp. 115-123. (1995)
41. Er, O., Cetin, O., Bascil, M.S., Temurtas, F.: A Comparative Study on Parkinson's Disease Diagnosis Using Neural Networks and Artificial Immune System. Journal of Medical Imaging and Health Informatics 6, 264-268 (2016)
42. Nasaroui, O., Gonzalez, F., Dasgupta, D.: The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling. In: Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on, pp. 711-716. IEEE, (2002)

43. de Castro, L.N., Timmis, J.: An artificial immune network for multimodal function optimization. In: Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, pp. 699-704. IEEE, (2002)
44. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. In: Genetic and Evolutionary Computation–GECCO 2004, pp. 287-298. Springer, (2004)
45. De Castro, L.N., Von Zuben, F.J.: The clonal selection algorithm with engineering applications. In: Proceedings of GECCO, pp. 36-39. (2000)
46. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. Artificial Immune Systems, pp. 153-167. Springer (2005)
47. McEwan, C., Hart, E.: On AIRS and clonal selection for machine learning. Artificial Immune Systems, pp. 67-79. Springer (2009)
48. Watkins, A.B.: AIRS: A resource limited artificial immune classifier. Mississippi State University (2001)
49. Burnet, F.M.: A modification of Jerne's theory of antibody production using the concept of clonal selection. Australian J. Sci. 20, 67-69 (1957)
50. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11, 10-18 (2009)
51. Stone, M.: Cross-validatory choice and assessment of statistical predictions. Journal of the royal statistical society. Series B (Methodological) 111-147 (1974)
52. De Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. Evolutionary Computation, IEEE Transactions on 6, 239-251 (2002)
53. Brownlee, J.: Immunos-81, the misunderstood artificial immune system. Faculty of Information & Communication Technologies (ICT) (2005)
54. Sudha, M., Selvarajan, S.: Feature Selection Based on Enhanced Cuckoo Search for Breast Cancer Classification in Mammogram Image. Circuits and Systems 7, 327 (2016)
55. Santiesteban, Y., Pons-Porrata, A.: LEX: a new algorithm for the calculus of typical testors. Mathematics Sciences Journal 21, 85-95 (2003)
56. Velayutham, C., Thangavel, K.: Unsupervised quick reduct algorithm using rough set theory. Journal of Electronic Science and Technology 9, 193-201 (2011)

An Experimental Comparison of Credit Risk Classification

Yosimar O. Serrano-Silva[1], Yenny Villuendas-Rey[2], Cornelio Yáñez-Márquez[1]

[1]Centro de Investigación en Computación del Instituto Politécnico Nacional, Avenida Juan de Dios Bátiz esq. Miguel Othón de Mendizábal, Nueva Industrial Vallejo, Gustavo A. Madero, CP 07738, CDMX, México.
[2]Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional, Av. Juan de Dios Bátiz s/n, Nueva Industrial Vallejo, Gustavo A. Madero, 07700 CDMX, México.

oswaldo17@live.com.mx, yenny.villuendas@gmail.com, coryanez@gmail.com.

Abstract. Credit is a fundamental aspect for finances, and there is the necessity of developing automated decision-making systems, which to some extent, can reduce the risk involved for the institutions granting credit. In this paper, we tested several supervised classification algorithms, and compare their performance over some well-known credit datasets, according to the Area under the ROC Curve.

Keywords: credit risk classification, supervised classification, imbalanced data.

1 Introduction

In commercial banking (business, personnel, etc.), Institutions assume some credit risk in every single asset operations that they perform (loans, lines of credit, guarantees, etc.) because on the one hand, this Institutions can never know everything about the customers and on the other, compliance with the payment obligations depends on events that nobody knows if could happened or not, that is, there is uncertainty about whether or not the customer will pay its debt. For those reasons, a lot of models have been proposed to evaluate credit risk [1].

In the literature, we can find a lot of different techniques that have been proposed to solve the problem of the credit risk and credit approval. Some approaches are based on the credit area, but address the issue of how credit managed badly can produce personal bankruptcy [2]. However other approaches try to solve some issues from credit rating systems using hidden Markov models [3] and others, based on the increasingly important role that social media has been playing sharing individual's opinions on many financial issues, analyzing whether these opinions can accurately predict credit risk[4].

Nevertheless uncertainty still exists in the financial area and there is the necessity of developing automated decision-making systems, which to some extent, can reduce the risk involved for the institutions granting credit or any of the operations previously mentioned.

Therefore, risk analysis, with all its factors and types, has some difficulties that nowadays we are still facing: How to integrate all these variables within an automated decision-making system? And after that, How to measure the significance of each variable and its contribution to the adoption of one decision or another?

In order to make a decision automatically, one solution is to use a model of pattern recognition to solve the classification task [5], i.e., from a certain dataset containing relevant information about customers that request a type of credit to a financial Institution, the classifier can help these Institutions to decide whether it is appropriate or not to grant the request.

However, one of the drawbacks when working with datasets that contain financial information is that these, in most cases, have missing values, unbalanced classes and have mixed attributes types[6], for which the classification model should be able to address this situation.

An unbalanced class is present in a dataset when one of the classes has more elements than the others. The problem working with this kind of datasets is that, in general, this situation creates biased learning. At the moment of testing this phenomenon used to give us inaccurate results about the performance of the classifiers due to the biased learning causes that the classifiers appropriately recognize only the elements of the ruling class.

In this article an experimental studio about the performance of different supervised classifiers with credit datasets is presented, which in most of the cases has missing values, mixed attribute types and unbalanced classes.

The rest of the paper is organized as follows. Section 2 describes in detail some aspects of the classifiers used in this comparison and section 3 offers a discussion about the results obtained. Finally the paper ends with some conclusions and future research suggestions.

2 Sampling and error measurement

For the purpose of validating the performance of a classifier the most common method applied in the literature is stratified cross-validation (SCV). This technique involves partitioning the dataset into two complementary subsets. The first subset is used for training the classifier and the second one is used for testing. It places an equal number of patterns from every class into each partition to keep the same class distributions.

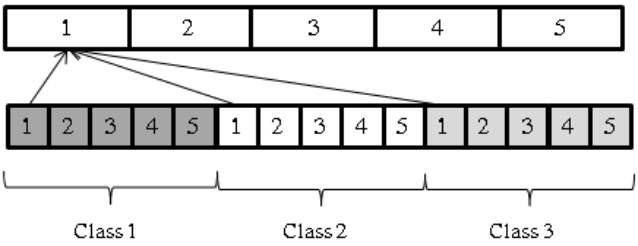


Fig. 1. Process to divide the dataset into k=5 subsets following the SCV technique.

To achieve that, as it shown in the Fig.1, firstly it is necessary to divide the complete data set into k different partitions, which in turn, are formed by one of the k partitions from the different classes.

After that, one of these subsets is taken as testing data and the remaining k-1 subsets are used as training data. This process is repeated k times and every subset is use as testing data exactly once.

As a result of the characteristics of the datasets from the financial and credit environment, it is necessary to choose a correct error measurement that can handle the problem of unbalanced classes and avoid inaccurate results. A metric that meet this requirement is the Area under the ROC curve (AUC) that is a popular classification metric which exhibits the benefit of being independent of the class distribution. The results of this measurement can be interpreted as follows: ideal classification model if the value of AUC is 1.0 and random classifier if the value obtained is 0.5. This measurement has been demonstrated that it can be calculated as the average of the True Positive Rate (TPR) and True negative Rate (TNR) for discrete classifiers by Sokolava et al.[7].

Table 1. Confusion matrix.

	Predicted as Positive	*Predicted as negative*
Positive instances	TP	FP
Negative instances	FN	TN

$$AUC = (TPR + TNR) / 2 \quad (1)$$

$$TPR = TP / (TP + TN) \quad (2)$$

$$TNR = TN / (TN + TP) \quad (3)$$

3 Results and discussion

3.1 Datasets

To carry out the different experiments, three datasets that belong to credit environments were used, which meet with the characteristics that are often present in this environment , it means that in the three datasets we can find missing values, data with hybrid types and unbalanced classes. These datasets were taken from the Machine Learning repository of the University of California [8].

Table 2. Caracteristics of the datasets used in this work.

Data set	*Instances*	*Attributes*	*Classes*	*Missing values*	*Unbalance Ratio*
Credit-australian	690	16	2	Yes	1.247
Credit-german	1000	21	2	No	2.333
Credit-approval	690	16	2	Yes	1.247

The German credit data corresponds to credit approvals. It has 1000 records with 20 attributes (7 numerical, 13 categorical) and do not have missing values. This dataset includes a cost matrix, due to the fact that it is considered worse to classify a client as good when is bad, than define a customer as bad when in fact is good.

The Credit Approval dataset contains 690 instances with 15 attributes (continuous and nominal) and presents some missing values. The purpose of this dataset is to predict whether an instance had a credit approved or not. The Australian credit dataset is a variation of the first one, used by the Statlog project [9].

3.2 Algorithms to compare.

Nearest Neighbor (1-NN).

Nearest Neighbor model [10] is part of the family of learning techniques called instance-based learning. The learning of this kind of algorithms is limited to stock in memory the patterns from the training set. This classifier is based on the idea that individuals from a population often share some similar properties and certain characteristics with the individuals around them. Thus, the classification of a pattern is carried out using the closest instances of the training set based on a dissimilarity measure. Due to the use of a distance measure, algorithms like this are called minimum distance classifiers.

One of the most popular similarity measures to numerical attributes is the Euclidean distance.

$$d(y,x) = \sqrt{\sum_{j=1}^{n}(y_j - x_j)^2} \quad (4)$$

C4.5

C4.5 algorithm builds decision trees from a dataset using the information entropy concept [11]. C4.5 chooses, at each node, the attribute of the pattern that splits effectively its set of samples into subsets improved in one class or the other. The criterion of splitting is the difference in entropy (normalized information gain). The attribute with the highest normalize information gain value is chosen to make the decision. Finally, this algorithm has three base cases:

- Instance of previously-unseen class encountered. The algorithm makes a decision node higher up the tree using the expected value.
- None of the features provide any information gain. Once more the algorithm makes a decision node higher up the tree using the expected value.
- The algorithm makes a decision node higher up the tree using the expected value. C4.5 creates a leaf node for the decision tree saying to choose the class.

Repeated Incremental Pruning to produce Error Reduction (RIPPER).

RIPPER is a classification algorithm that was proposed by William W Cohen [12]. It is based on association rules with reduced error pruning (REP), which is a common and effective technique of decision tree algorithms. In REP for rules algorithms, the training data is split into a growing set and a pruning set. First using some heuristic method, an initial rule set is formed (growing set). This overlarge rule set is then repeatedly simplified by applying one of a set of pruning operators typical pruning operators would be to delete any single condition or any single rule. At each stage of simplification, the pruning operator chosen is the one that yields the greatest reduction of error on the pruning set. Simplification ends when applying any pruning operator would increase error on the pruning set [12].

Multilayer Perceptron (MLP).

The Artificial Neural Network is a learning paradigm based on biological neural networks, in particular the human brain. Anatomically this system is composed for networks of biological neurons interconnected, which are able to process and conduct electrical impulses to produce an output. In 1943 it was proposed an abstract and simple model of an artificial neuron as a binary device [13]. This model has an operating threshold below which this neuron is inactive. Also, it has excitatory and inhibitory inputs, and depending on if there is any of these inputs the neuron is active.

This model is very simple, if there is not an inhibitory input, the resultant of the excitatory inputs is determined and if this is greater than the threshold, the output is 1 otherwise is 0 (Fig.2).

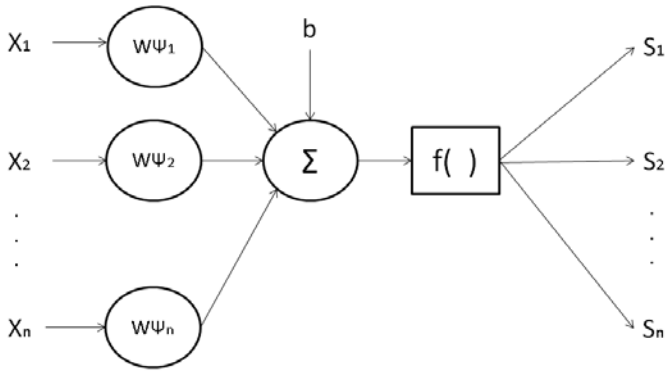


Fig. 2. Artificial neuron scheme.

Based on the work of McCulloch and Pitts, in 1953 it was proposed the perceptron [14]. One of the most interesting characteristics of this model was its ability of learning to recognize and classify objects. The perceptron was constituted by a set of input sensors which receives the patterns to recognize or classify and am output neuron to do the classification task. Nevertheless, this model was not capable to converge on good solutions in problems with classes linearly non-separable.[15].

Finally in 1986 the Multilayer Perceptron (MLP) [16] was proposed to solve the limitations of the perceptron. This network consists of multiple layers of artificial neurons; the most common architecture of a simple MLP network has 3 layers: an input and an output layer with one hidden layer however, the general model allows use an unlimited number of hidden layers.

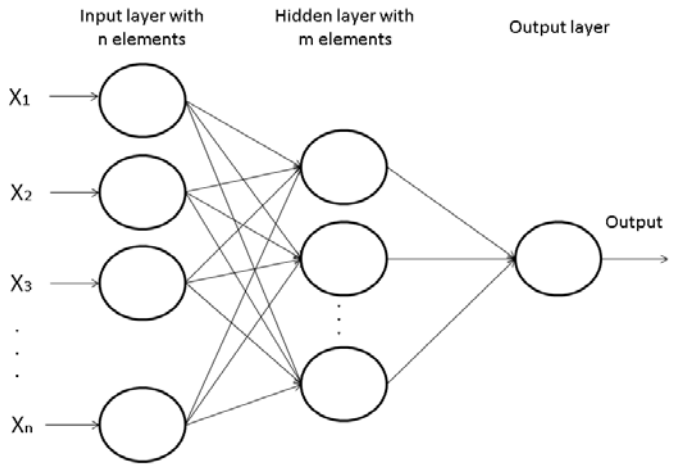


Fig. 3. General model of a MLP network with one hidden layer

Finally the supervised training stage is one of the most popular algorithms called back-propagation. The bases of this algorithm are in the error-correction learning rule [16].

Sequential Minimal Optimization Algorithm for training a Support Vector classifier (SMO).

Sequential Minimal Optimization (SMO) [17] is an algorithm for training Support Vector Machines [18] and was proposed to solve the problem of the very large quadratic programming optimization problem that implies this kind of training.

Considering a classification problem with a dataset (x_1, y_1),…, (x_n, y_n) where x_i is an input vector and y_i is a binary label corresponding to it. A soft-margin support vector machine is trained by solving a quadratic programming problem, which is expressed in the dual form as follows:

$$\underset{\infty}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j K(x_i, x_j) \alpha_i \alpha_j, \tag{5}$$

Subject to:

$$0 \leq \alpha_i \leq C, \quad for\ i = 1, 2, \ldots, n, \tag{6}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{7}$$

Where C is an SVM hyperparameter and $K(x_i, x_j)$ is the kernel function, both supplied by the user; and the variables α_i are Lagrange multipliers.

This is an iterative algorithm to solve the optimization problem. SMO converts this problem into a set of smallest possible sub-problems, which are then solved analytically. Due to the fact of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2 the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C \tag{8}$$

$$y_1 \alpha_1 + y_2 \alpha_2 = k \tag{9}$$

And this reduced problem can be solved analytically. The algorithm proceeds as follows [17]:

- Find a Lagrange multiplier α_1 that violates the Karush–Kuhn–Tucker (KKT) conditions for the optimization problem.
- Pick a second multiplier \alpha_2 and optimize the pair (α_1, α_2).
- Repeat steps 1 and 2 until convergence.

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved.

Naive Bayes (NB).

Naïve Bayes algorithm [19] assumes, for an instance x that its attributes x1, x2, …, xn have a conditional independence due to its class. For this reason the conditional likelihood of every attribute can be expressed as follows.

$$p(x|\omega_i) = \prod_{j=1}^{n} p(x_j|\omega_i) \quad (10)$$

Using the Bayes theorem, the posteriori likelihood is:

$$p(\omega_i|x) = \mathrm{p}(\omega_i) \prod_{j=1}^{n} p(x_j|\omega_i) \quad (11)$$

Finally, for every pattern of the testing set is given a class as is describe in the following equation

$$\omega^* = argmax_{\omega j} \mathrm{p}(\omega_i) \prod_{j=1}^{n} p(x_i|\omega_j) \quad (12)$$

Each was tested with the different datasets in Waikato Environment for Knowledge Analysis (WEKA) software [20] in its version number 3.6.13 using the default parameters offered.

3.3 Error Measurement.

The results obtained with the different models to every dataset, using the Stratified Cross Validation with k=5 as model validation technique, are shown in Table 2. We use the Area under Roc curve (AUC) [7] as performance measure.

Table 3. Area under de curve ROC

Classifiers	*Credit-australian*	*Credit-german*	*Credit-approval*
1-NN	0.8110	**0.6855**	0.8110
C4.5	0.8530	0.6540	0.8530
RIPPER	**0.8605**	0.5985	**0.8605**
MLP	0.8390	0.6695	0.8390
SMO	0.8580	**0.6855**	0.8580
Naive Bayes	0.7595	0.6785	0.7595

Conclusions and future work

Nowadays there are a lot of datasets from the financial environment that are very important to the different automated decision-making systems, but these datasets have some characteristics that make this task more complicated. In this paper we compared six different classification techniques in credit environment: Nearest Neighbor, C4.5,

Repeated Incremental Pruning to produce Error Reduction, Multilayer Perceptron, and Sequential Minimal Optimization Algorithm for training a Support Vector classifier and Naive Bayes.

These techniques were compared by using the Area under the curve ROC due to the problem of the unbalanced classes present in these credit datasets. Our studies showed that SMO model turned out to be best classifier for Credit-Australian and Credit-approval dataset, but in the Credit-German dataset both 1-NN and SMO share the best performance.

Finally an analysis of statistical significance on classifiers that were compared, was not possible because the number of datasets was not enough to carry out this kind of analysis.

Acknowledgments

The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Investigación en Computación, and Centro de Innovación y Desarrollo Tecnológico en Cómputo), the Consejo Nacional de Ciencia y Tecnología, and Sistema Nacional de Investigadores for their economical support to develop this work.

References

1. Li, Y., Zhou, Z.: Research on Model for Evaluating Risks of Venture Capital Projects. J. Risk Anal. Cris. Response. 1, 142–148 (2011).
2. Xiong, T., Wang, S., Mayers, A., Monga, E.: Personal bankruptcy prediction by mining credit card data. Expert Syst. Appl. 40, 665–676 (2013).
3. Petropoulos, A., Chatzis, S.P., Xanthopoulos, S.: A novel corporate credit rating system based on Student's-t hidden Markov models. Expert Syst. Appl. 53, 87–105 (2016).
4. Yang, Y., Gu, J., Zhou, Z.: Credit risk evaluation based on social media. Environ. Res. 148, 582–585 (2015).
5. Alickovic, E., Subasi, A.: Medical Decision Support System for Diagnosis of Heart Arrhythmia using DWT and Random Forests Classifier. J. Med. Syst. 40, 1–12 (2016).
6. Abdi, L., Hashemi, S.: To combat multi-class imbalanced problems by means of over-sampling and boosting techniques. Soft Comput. 19, 3369–3385 (2015).
7. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar, A. and Kang, B. (eds.) AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings. pp. 1015–1021. Springer Berlin Heidelberg, Berlin, Heidelberg (2006).
8. Lichman, M.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml.
9. Michie, E.D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning , Neural and Statistical Classification. (1994).

10. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. 13, 21–27 (1967).
11. Salzberg, S.L.: C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. Mach. Learn. 16, 235–240 (1994).
12. Cohen, W.: Fast effective rule induction. Twelfth Int. Conf. Mach. Learn. 115–123 (1995).
13. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133 (1943).
14. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. 65, 386–408 (1958).
15. Minsky, M.L., A. Papert, S.: Perceptrons. MIT Press (1969).
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by backpropagation error. Nature. 323, 533–536 (1986).
17. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. 1–21 (1998).
18. Cortes, C., Vapnik, V.: Support-Vector Networks. Mach. Learn. 20, 273–297 (1995).
19. Duda, R., Hart, P., Stork, D.: Pattern Classification and Scene Analysis. (1973).
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. 11, 10–18 (2009).

Control system for automatic positioning of a satellite antenna

Edgar Roberto Ramos Silvestre, Eynar Calle Viles, Isabel Morales Ledezma

Universidad Privada del Valle UNIVALLE., Bolivia
eramoss@univalle.edu ; eynarcalle@gmail.com; Isabel.morales.le@gmail.com

Abstract. This work proposes a control system for automatic positioning of a satellite antenna. The mathematical model of a DC motor is explained and the simulation of the entire system is demonstrated. Likewise, is mentioned the control method used in the system and the different reactions of the prototype. Also, the electronics modules and hardware modules are explained. Finally, the test acquired with the prototype is shown with a DVB S-2 system. The entire simulation software is developed in Matlab, a portable software designed to perform different parameters such as elevation, azimuth and polarization LNB for a satellite link.

Keywords: satellite link, elevation, azimuth, DVB S-2.

1 Introduction

Nowadays, satellites play an important role in global communication including phones, data networks, video streaming and transportation, as well as the television and radio diffusion straight to the user [1].

Satellite communication offers significant advantages over other types of long-distance communication. To mention some, the ability to communicate between two or more points at considerable distances; the ability to disseminate and collect signals on any ground surface with direct line of sight; and the ability to transport services to remote regions without using wired media, where point to point connection would not be practical to implement the links.

Satellites are used extensively for communication purposes as well as in navigation systems, scientific research, data capture remotely, military reconnaissance, detection of natural disasters, tele-education, tele-health or any other application. All these applications require one or more direct communication links with the satellite [2].

Moreover, it is observed that control earth stations of great magnitude, have a system of motors to control a satellite dish. So from the beginning of the appearance of these electric actuators, there is a need to control its position, speed and acceleration for generating a controlled motion, in mechanisms [4].

The use of electric motors has led to the use and development of control schemes from the simplest to those considered modern control techniques using feedback

sensors for measurements of variables associated with the system. Constantly aiming to improve the performance of engines, new topologies and control methods are developed.
For correct alignment of an antenna with a satellite three movements called azimuth, elevation and polarization are required. The first two will be of interest for the project to control the movement of the antenna. In this article a simple ON / OFF control and easy to handle, with respect to a PID controller is proposed.

2 Mathematical model and control law

2.1 Mathematical model of a DC motor

In Figure 1, a DC motor moves a load through a box of gears. This load represents a parabolic antenna which obtains geostationary satellite signal through movements in azimuth and elevation. To perform the movement of azimuth on the satellite dish, a DC motor is used; and another for movement in elevation. Thus, it can be controlled an earth station.

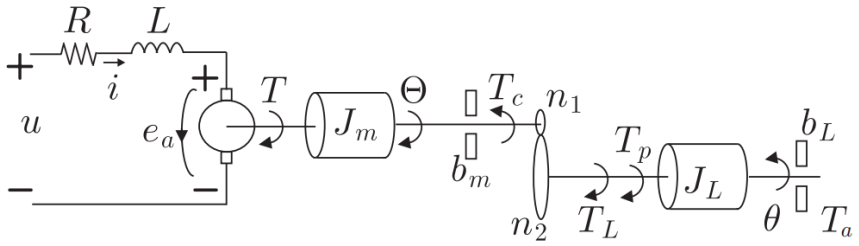


Fig. 1. DC motor, which drives a load through gears n1 and n2 where n2 represents the support of the parabolic antenna, a gear with 86 teeth; and n1 represents the motor's gear that corresponds to an endless screw.

The mathematical model of the DC motor is resumed from [5] where an electrical and mechanical subsystem is presented.

Electrical subsystem model

When applying Kirchhoff's Voltage Law to the armature circuit that is shown in Figure 1, it results:

$$applied\ voltage = \sum brownouts\ in\ the\ grid$$

$$u = voltage\ drop\ across\ the\ inductor + voltage\ drop\ across\ the\ resistor + counter\ electromotive\ force$$

$$u = L\frac{di}{dt} + R\ i + e_a$$
$$e_a = k_e\ \dot{\Theta} \quad (1)$$

Where "$\dot{\Theta}$" represents the first time derivative of the variable.

Mechanical subsystem model of the DC motor

In this sector, it should be applied the second law of Newton. As there are two different bodies, Newton's second law should apply to each of these bodies separately.

Rotor model:

$$Inertia * angular\ acceleration = \sum Torque\ J_m$$

$$J_m\ddot{\Theta} = torque\ generated - friction - load\ torque.$$

$$J_m\ddot{\Theta} = T - b_m\dot{\Theta} - T_c$$
$$T = k_m i \quad (2)$$

Load model:

Before applying Newton's second law to the load some important relationships, motor gears and gears of the load are determined,

$$n = \frac{n2}{n1}\ ; \qquad T_L = nT_c\ ;\ \ \Theta = n\theta \quad (3)$$

Now it can be applied Newton's second law to the load:

$$J_L\ddot{\theta} = T_L - b_L\dot{\theta} \quad (4)$$

From (1) and (2) - (3), the combined model of the DC motor and the load is obtained:

$$J_m\ddot{\Theta} = k_m i - b_m\dot{\Theta} - \frac{1}{n}T_L$$
$$J_m\ddot{\Theta} = k_m i - b_m\dot{\Theta} - \frac{1}{n}\left(J_L\ddot{\theta} + b_L\dot{\theta}\right)$$

$$J_m \ddot{\theta}\, n = k_m i - b_m \dot{\theta} n - \frac{1}{n}\left(J_L \ddot{\theta} + b_L \dot{\theta}\right)$$
$$(J_m n^2 + J_L)\ddot{\theta} + (n^2 b_m + b_L)\dot{\theta} = n\, k_m i \quad (5)$$

If is defined:

$$J = n^2 J_m + J_L \qquad b = n^2 b_m + b_L \quad (6)$$

Then:

$$J\ddot{\theta} + b\dot{\theta} = n\, k_m i \quad (7)$$

Finally, the mathematical model of the DC motor is given by (1) and (7), which expressed in terms of axis angular velocity $\varpi = \dot{\theta}$ are defined as:

$$L\frac{di}{dt} = u - R\, i - k_e\, n\, \varpi$$

$$J\frac{d\varpi}{dt} = -b\, \varpi + n\, k_m i \quad (8)$$

2.2 Control law

The evolution of modern storage electronic equipment and processing of data, enables the improvement in control systems. Thus, they can be implemented in microcontrollers for motor control. This is very important in any mechatronic system, and imposes challenges that defy designers based on control theory controls. Traditionally motors were controlled manually, the modern control came with semiconductors, subsequent developments in power electronics and microelectronics allowed the development of better drivers with high performance and cheaper components.

Currently, new electronic devices, microprocessors, microcontrollers and digital signal processors are developed, which allow the evolution of more sophisticated and economic controls.

In this article a simple and economical ON / OFF control, implemented in a microcontroller ATMEGA 328 is proposed, which is simulated in Matlab Simulink based on the mathematical model of the DC motor. Figure 2 shows that the ON / OFF control is connected to the DC motor and feeds back the position variable of the DC motor controller. In addition, it can be implemented with low cost electronics covering all the needs of the system.

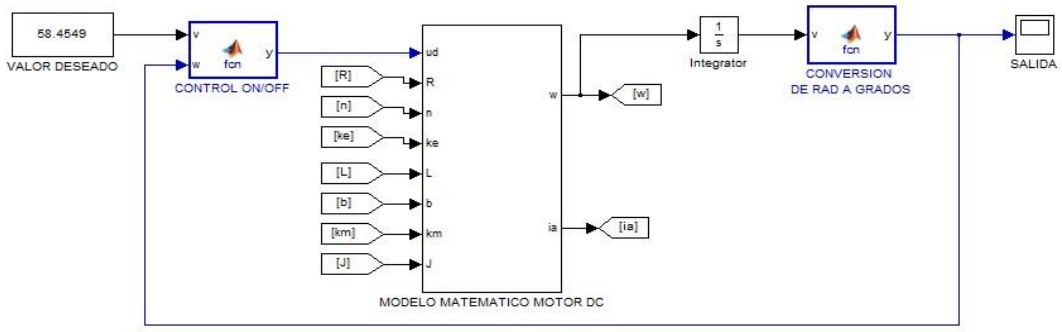


Fig. 2. Design of a DC motor controller, using an ON/ OFF controller which positions the system in the required direction.

2.3 Simulation

Once the mathematical model of the DC motor is known and the parameters involved as: resistance R, inductance L, viscous friction b_m, rotor inertia J_m, gear ratio n, back electromotive force k_(e)and constant torque motor k_m, it are parameters characterizing a DC motor, a series of experiments performed allowed the observation of the different reactions of the DC motor. In Figure 3, the parameters used to perform the simulation are shown as well as the DC motor plotting of equation (8) obtained from the mathematical model of the DC motor.

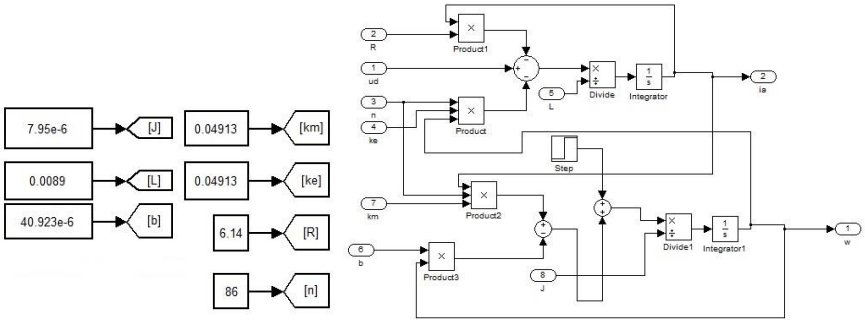


Fig. 3. (a) DC motor parameters, these parameters allow the DC motor recreate in a simulation environment, (b) Matlab-Simulink representation of the mathematical model of the DC motor.

Matlab is a working environment for scientific computing with a Simulink extension that is used to model, simulate and analyze localization system [6].

In Figure 4, the interface of simulation designed is observed; theoretical and practical tests were performed with the Bolivian satellite Tupac Katari that allowed to achieve and analyze the different reactions of the system.

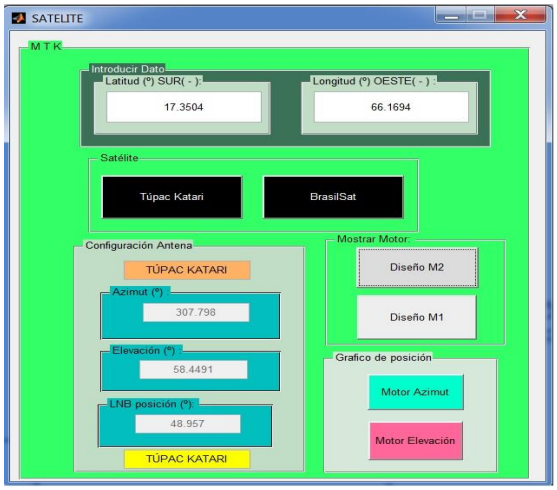


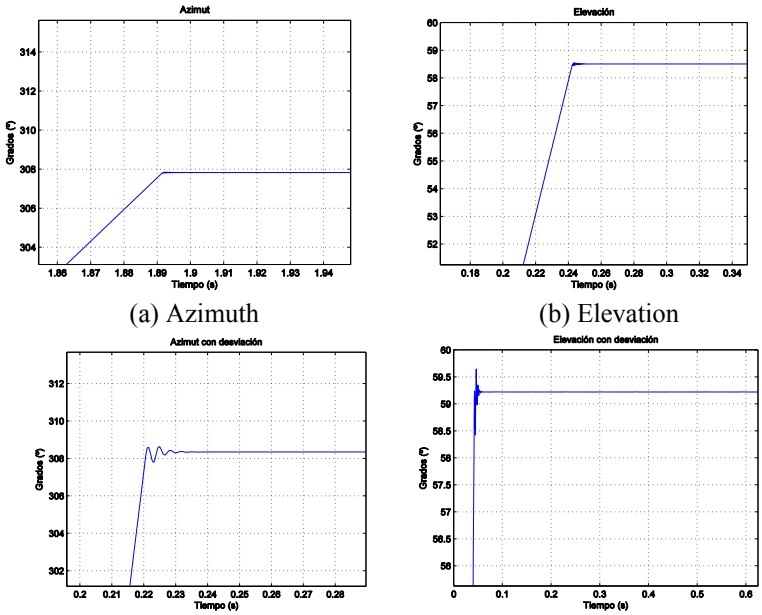
Fig. 4. Simulation software has a graphical user interface (GUI), where it can graphically analyze the results.

When entering data as latitude and longitude position, it is possible to determine the elevation and azimuth that are required for a satellite link. With the simulation software it is possible to calculate these parameters.

With the mathematical model of the DC motor, using standard values as shown in Figure 3, you can get the simulation of a DC motor by pressing the "Motor azimut" (motor azimuth) button and "Motor elevación" (motor lift). In Figure 4, the system response is obtained.

Figure 5, shows the response of the DC motor in azimuth, it is observed the position acquired by the DC motor after being controlled. To achieve this response in azimuth, was used n2=86 and n1=1. Where n1 represents an endless screw and n2, a 86 teeth gear. For testing in elevation, was used n1=1 and n2=58, so that n1 also represents an endless screw, and n2, a 58 teeth gear.

In the simulation it is required to position the elevation system with 307.79 degrees and an azimuth of 58.45 degrees respectively. In Figure 5, (a) and (b) show that the system is positioned in the calculated data, but a variation is observed in the system (c). If the values of n1 and n2 are changed, the system responds differently. Was achieved the variation n = 20 according to equation (3), where also was determined that increasing the number of teeth n1, the system becomes unstable and introduces error in the final result.



(a) Azimuth (b) Elevation

(c) Azimuth deviation (d) Elevation deviation

Fig. 5. (a) DC motor response in azimuth, (b) DC motor response in elevation, (c) DC motor response with azimuth deviation, (d) DC motor response with elevation deviation.

3 Mechanics structure and hardware

3.1 Structure of satellite antenna

To build the prototype, the pieces were designed with SolidWorks software. A satellite dish offset type was used with: 65cm a major axis and a minor axis of 60 cm, an angle offset 22.61º, a gain of 36.65db Ku-band reflector to 12.5GHz and an efficiency of 75% [3].

It is necessary that the structure is well designed so not to have variations when the system is running. Consequently, for the development of each of the prototype parts, rigid materials were used.

Figure 6, shows the design of the main axis corresponding to the azimuth prototype, this piece was made to fit and with plastic material. Contains a gear (for simulation corresponds to the azimuth motor, n2) of 86 teeth which allows to obtain precision when the prototype is operating.

Also, the DC motor with an incremental encoder adapted to an endless screw is observed (for simulation corresponds to n1). This screw was produced to fit as and in plastic material.

Fig. 6. (a) DC motor with endless screw, equivalent to n1=1, (b) Principal axis equals n2 = 86. The DC motor uses an incremental encoder, resolution 400 pulses.

By considering that the worm is equivalent to a gear having a single tooth, this helps in system accuracy. It also generates a resting state system, preventing the main axis to move when this is out of operation for any disturbance.

3.2 Electronic boards

Three electronic modules were implemented, each controlled by a development board that works as a master. Each module, performs work independently so that the system can perform multiple processes using an integrated development environment.
Each of the electronic modules are connected by a i2c interface. This allows independent control and perform multiple processes, each module has its own address so it can be identified in the system.
The master controller is an Arduino Uno, after being processed by the electronic modules, is responsible for performing queries and receive information. Furthermore, the designs of these three electronic modules are shown in Figure 7. They were developed independently, so that each module corresponds to a printed circuit board that can function as an expansion, this allows the system to be scalable.

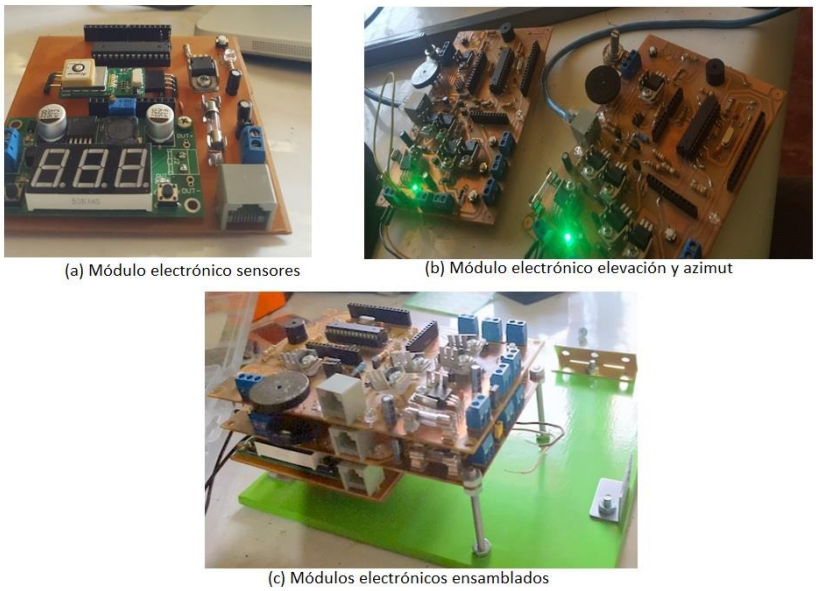


Fig. 7. (a) Electronic sensor module, is responsible for feeding the modules that are connected, as well as support to the sensors 10DOF IMU and GPS. (b) Electronic module elevation and azimuth, is responsible for performing the position control of DC motor, using microcontroller ATMEGA 328p. (c) Assembled electronic modules.

Using Microsoft Visual Studio, integrated development environment for Windows operating systems, a program is developed to be run as an application with a user interface on a computer.
When entering latitude and longitude parameters, the necessary data are calculated: elevation, azimuth and angle of the LNB to direct a parabolic antenna to the satellite Túpac Katari.
Likewise, it can connect with the prototype and get the latitude and longitude parameters acquired by the GPS, and achieve calculate the elevation data, azimuth and angle of the LNB in the position where the prototype is located.
The application contains a graphical interface as shown in Figure 8, where can be observed the values calculated with data of latitude (-17.3504) and longitude (-66.1694), different tests were performed at place.

3.3 Software

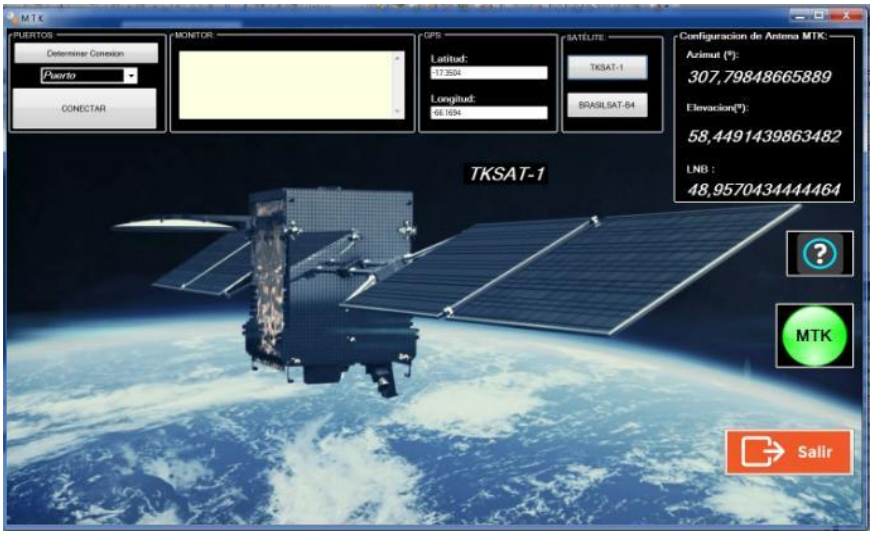


Fig. 8. Software user interface, application to calculate parameters of azimuth, elevation and polarization of LNB.

4 Results

A variety of tests were conducted in Bolivia. Especially in different parts of the city of Cochabamba, these test points correspond to:

- Point 1 Latitude -17.3504 Longitude -66.1694 (Cochabamba, North Zone)
- Point 2 Latitude -17.3303 Longitude -66.2257 (Cochabamba, Tiquipaya, Universidad del Valle)
- Point 3 Latitude -17.4059 Longitude -66.0333 (Cochabamba, Sacaba)

Among the tests, it is accomplished a signal strength of 84%, and a signal quality of 77%. In addition, it manages to capture 20 FTA channels from satellite Túpac Katari. In Figure 9, the result of the tests is shown. It is observed that the test performed number 197 corresponds to 317° azimuth antenna, maximum values acquired by satellite link. In addition, is detected ± 4 degrees for fine adjustment, this due to the radiation power of the satellite on Bolivian territory, 54.5dbW that allows the system to acquire the signal at any position within this range, to make an adjustment.
In Figure 10, the final prototype stands out, performing tests in 2. The prototype is fully portable with economically affordable and easily accessible materials in the local market.

Decima Prueba (pos2)			
Prueba	**Grados (º)**	**Intensidad de Señal (%)**	**Calidad de Señal (%)**
190	310	44	5
191	311	44	5
192	312	44	5
193	313	87	58
194	314	87	71
195	315	87	71
196	316	84	76
197	317	84	77
198	318	84	76
199	319	84	76
200	320	85	74
201	321	86	71
202	322	44	7
203	323	44	5
204	324	44	5
205	325	44	5
206	326	44	5
207	327	44	5
208	328	44	5
209	329	44	5
210	330	44	5

Fig. 9. Results Table, in the test number 197 is possible to obtain a signal strength of 84% (intensidad de señal), and a signal quality of 77% (calidad de señal).

Fig. 10. Final prototype, performing tests with a DVB-S2 receiver with a frequency of 11670 MHZ, symbol rate of 150000 Ksps and vertical polarization.

5 Conclusions

It was experienced automatic control laws on DC motors in order to obtain the correct position of azimuth and elevation with a satellite dish. In addition, it was

reached to make a satellite link with 20 free channels Tupac Katari satellite from Bolivian property.

It was also possible to design electronic and software modules necessary for geostationary satellite location of the Tupac Katari. The final prototype with all modules was implemented by testing and calibration of the system in three locations corresponding to Tiquipaya, Zona Norte and Sacaba in Cochabamba, Bolivia.

A simulation software was performed to observe the reactions of the controller, as well as the entire system. Furthermore, an application was developed to interact with the user.

By performing several tests with the prototype implemented, it was possible to note a variation of 8 degrees, to fine-tune. In the same way, it was possible to observe that using an incremental encoder of 400 pulses per revolution in a dc motor with an endless screw and a 86 teeth gear: got higher resolution and stability in the project. That is, 34,400 pulses per revolution (360 ° 0 °) therefore, the result is more accurate.

It is possible to obtain Túpac Katari satellite signal with a DVB-S2 receiver, to appreciate free channels.

An installation of an offset type Ku-band parabolic satellite may take several hours to even days, if not aware of essential parameters to perform an installation. This project, reduces installation time, providing ease and portability when is time to make an installation.

6 References

1. Gérard Maral, M. B. (2009). *Satellite Communications Systems, Techniques and Technology* (quinta ed.). John Wiley & Sons Ltd.
2. Justino Ribeiro, J. A. (2008). *Propagación de ondas electromagnéticas,principios y aplicaciones.* São Paulo: Érica.
3. Motta Marins, C. N. (2004). *Estudo analítico e numérico de um enlace digital de comunicaçao via satélite em condiçao orbital geoestacionária.* Santa Rita do Sapucaí.
4. Tafoya Sánchez, J. J. (2010). *Control de velocidad angular de motores de corriente directa mediante ténicas de control automático.* Mexico, D.F.
5. Orbegoso Guerrero, A., Muños Villalobos, C., & Villalta Ramirez, A. (2010). *Software para ciencia e ingeniería MATLAB.* Lima-Perú: Macro E.I.R.L.
6. Justino Ribeiro, J. A. (2012).*ENGENHARIA DE ANTENAS, Fundamentos, Projetos e Aplicaçoes.* Sâo Paulo: Érica.

Electrical Impedance Tomography image reconstruction using backprojection with OpenCV

Miguel Ángel San-Pablo-Juárez[1], Eduardo Morales-Sánchez[1], Fernando Ireta-Moreno[2], Raúl Alejandro Ávalos-Zúñiga[1], and José Joel González-Barbosa[1]

[1] Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada unidad Querétaro, Instituto Politécnico Nacional
miguelangel.sanpablo@gmail.com
[2] División de Ingenierías campus Irapuato-Salamanca, Universidad de Guanajuato
fireta@ugto.mx

Abstract. In this work is presented a graphical user interface implementation for imaging Electrical Impedance Tomography by Sheffield backprojection. C language is used and the OpenCV library is implemented to display a tomographic image with color levels according to the conductivity level in the studied medium. This implementation is presented as an alternative for electrical impedance tomography imaging using filters and colormaps without Matlab software. Some real data reconstructions were made to validate the correct operation and the resolution of the image was improved using filters.

Keywords: Electrical Impedance Tomography, OpenCV, Image reconstruction, Sheffield backprojection

1 Introduction

Actually the Electrical Impedance Tomography image reconstruction is made with a specialized software called EIDORS: *Electrical Impedance Tomography and Diffuse Optical Tomography Reconstruction Software* that is free and open source. Despite these advantages, it is required to have installed a Matlab version (or Octave) to use all its features. The hardware that collects the measured data to reconstruct a tomographic image can be adapted to Matlab, and it is possible to generate a data output necessary to image a conductivity distribution. Nowadays, the most used output is an image formed by a finite number of pixels. A compiled specific software for some specific application of Electrical Impedance Tomography (EIT) is more practical because in this way the involved algorithms that solve direct and inverse problems can be easily manipulated from code. As a

result it is possible to access to numerical values of conductivity or other variables of interest. This paper seeks to deploy tomographic images in a user-friendly interface through an OpenCV library as an easy and highly efficiently alternative. A colormap that gives contrast is displayed to differentiate conductivity values too.

1.1 Electrical Impedance Tomography

Electrical Impedance Tomography (EIT) is a technique that can produce images distribution of admitivity, or electrical conductivity, of electrically conductive objects by injecting known amounts of current, and measuring the resultant electric field at the surface of the object [1,2].

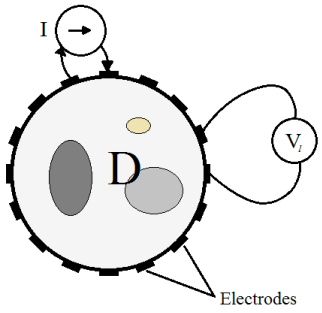


Fig. 1: Configuration of the current injection and potential measurement in EIT

In Fig. 1 is shown the typical configuration for the reconstruction problem in two dimensions, it consist of a 16 electrodes array placed on the surface of the region D. A pair of electrodes is used to inject a sinusoidal current I inside the object; the resulting potential distribution is measured pair by pair in the remaining electrodes. These measured voltages are denoted as V_l. After this is done, the procedure is repeated by injecting the current to the next pair of electrodes and measuring the new potentials. In this way,16 times the current injection it will be made until to complete a round, thus having a vector of 208 measurements potential difference between the electrodes on the surface of the object. A vector of measures, corresponding to 16 measurements of 13 potential values between the electrodes attached to the surface D of the object being studied, will have stored.

The governing equation in EIT is

$$\nabla \cdot (\sigma \nabla \phi) = 0 \in D \tag{1}$$

where σ represents the conductivity and ϕ represents the potential inside the region D.

This is the Laplace equation for the linear case, a second order partial differential equation and elliptical. For this two dimensional problem only the coordinates x and y are considered.

This equation models the flow of the electric current across the medium in region D, as it represents the divergence of the gradient for the potential ϕ. In this particular case, it measures the amount of electrical current flowing in and out of the region D in a circular form, typically used in EIT.

In another particular case, when the Laplace equation has conductivity $\sigma = k$, with $k = 1$ or a constant we are treating with an homogeneous medium where the same amount of current into one of the electrodes is exactly the same coming out, fulfilling the same but for a nonlinear governing equation case when using Poisson equation is modeled.

1.2 Polar and adjacent current injection

Two of the most commonly used methods for injecting current in EIT are the adjacent method method and the polar injection. The adjacent injection method applies the current between two adjacent electrodes (neighbor electrodes), and the potentials are measured in the remaining adjacent pairs of electrodes. This process is repeated until each one of the possible adjacent pairs have had an injection current [3]. This method is too sensible to conductivity contrasts near the boundary, and it is insensible to central contrasts. The polar injection (opposite electrodes), mainly used in brain EIT, applies a current to a pair of electrodes that are opposite 180 each other while the potentials are measured in the remaining pairs of electrodes. Polar injection strategy suffers the disadvantage that for the same number of electrodes, the available number of current injections that can be applied is lower than in the adjacent strategy. In this work is used the adjacent type current injection, since a total of 208 measurements of potential are required to enter the program and use the backprojection.

1.3 EIT image reconstruction

Electrical Impedance Tomography image reconstruction has been studied with static and differential techniques. Static reconstruction use a data set to carry out the reconstruction while differential reconstruction uses two data sets and computes a conductivity difference [4]. There are different reconstruction techniques, as the linear backprojection, Newton-Raphson method, Graz Consensus Reconstruction Algorithm (GREIT) [5,6] and the conjugate gradient method [7] to name the main. In this paper is used the linear backprojection technique by equipotential lines, also called Sheffield backprojection [8], which is a type of differential reconstruction. Several of these methods as well as solvers of the forward problem are included in the EIDORS software [9].

1.4 Linear Backprojection

The backprojection method is a reconstruction method widely used for imaging EIT. In Electrical Impedance Tomography, it is required to make a modification of the conventional backprojection in order to use this technique. In EIT the straight lines strategy cannot be used because changes in all elements of the object affect the measurements. After all current patterns are applied, the projected values are summed to obtain a value for a pixel. The algorithm is then [10]

$$\frac{\delta\sigma}{\sigma} = B\frac{\delta V}{V} \tag{2}$$

where B is the backprojection matrix. The lines used for the backprojection are the equipotential lines in each current injection since for the given element, the maximum sensitivity measured can be found at the intersection of the boundary and the line from the element. The equipotential lines are unknown due to the unknown conductivity, therefore they must be approximated. Usually the approach is based on the assumption that the object is circular, and it has a constant conductivity. There are different versions of the backprojection algorithm, one of the most used is the proposed by [11]. In this work is used the Sheffield backprojection matrix [12]. The reconstruction by backprojection presents a fast computing, since only multiplications are processed. The current injection technique for this case will be by adjacent electrodes.

2 Methodology

The methodology to obtain the internal conductivity of a region using EIT is proposed as follows:

First an electric current is injected and measurements are taken by some method, such as adjacent injection, readings of potential are taken between the remaining pairs of electrodes and the lectures are stored in a vector or a matrix for later use. Once the measured potential values are obtained, these must be entered into the program for the processing; as backprojection is used, it is necessary to have potential measurements made in a homogeneous medium and also measurements in a disturbed medium. In this case the measurements of the disturbed medium are the real measurements from the measured region. Subsequently backprojection is applied to obtain a vector that represents the conductivity values σ of finite small elements within the studied region. A distribution which can be represented by an image of pixels with a given color scale values of conductivity is obtained.

For the first stage, measured data is simulated with the potential values computed in the most widely used software for EIT called EIDORS (Electrical Impedance Tomography and Diffuse Optical Tomography Reconstruction Software [4]), if desired, you can also to perform the current injection via hardware,

but in this case only examples with preset values are explained. For the second stage the backprojection matrix B is used, whose values have been calculated by [12]. The implementation of the backprojection matrix in the program is done in C language to reconstruct the distribution of conductivity values according to the equation (2). For this, a dialog box is implemented in *Windows Forms* from Visual Studio and a tool type *PictureBox* is added. Behind this object, the presentation of the conductivity values calculated in backprojection will be programed. OpenCV is used to generate a gray scale or a list of colors for the color generation. The shape of quadrilaterals, the size and the outline of each of these parts are also considered tracing quadrilaterals that represents a matrix of pixels. First the deployment of a reconstructed gray scale image is made and finally an example is displayed with colors. The used colormap can be variable, being a standard the gray scale and the colormap configuration for *Jet*. Finally an image (32×32 pixels of low resolution) that can be filtered by the same OpenCV library is generated, and for this work a Gaussian filter is used.

3 Results

The graphical user interface was created using a Windows Forms application with Microsoft Visual Studio, in this part the operations relating to the equation (2) are programmed. First is calculated

$$V_e = \frac{V_{meas} - V_{ref}}{V_{ref}} \tag{3}$$

that computes the operation for obtaining $V_e = \frac{\delta V}{V}$, and thus obtain [11]

$$\sigma_e = B_{_M} \cdot V_e \tag{4}$$

where $\sigma_e = \frac{\delta \sigma}{\sigma}$. The linear system that is typed into the compilator is then represented as

$$[\sigma_e]_{912\times 1} = [B_{_M}]_{912\times 208}[V_e]_{208\times 1} \tag{5}$$

and σ_e represents a conductivity vector containing the values of each discrete element in which the measured region is divided, $B_{_M}$ is the backprojection matrix and V_e is the measured potential between electrode pairs in all projections by adjacent injection current.

For the graphics, a total of 16×13 potential measurements will be taken, and with the matrix $B_{_M}$ a total of 912 pixels will be taken with conductivity values represented by a color value. To adapt the pixels to a circular image, some values are ignored and are deployed only 856 values in the computer screen.

To display an image, first a homogeneous medium is selected, the values of potential are calculated by solving the forward problem in EIT and potential

values are stored; after, a medium is taken with an object with known conductivity within. In this case, a phantom was taken from the software EIDORS [13]. In the Fig. 2 is shown an example from EIDORS, where two objects are known, the first one with a conductivity value of 1.1 (*left perturbation*) and 0.9 (*right perturbation*). This software allows to show the conductivity values computed after to solve the inverse problem in EIT and the Fig. 2 shows the output in the right part. Once the solution for calculations of $\delta\sigma$ in C language

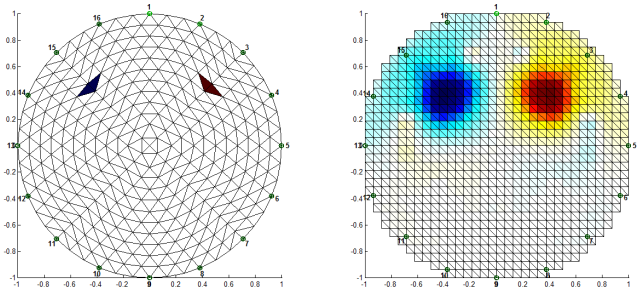


Fig. 2: Example with two perturbations inside a circular region and the conductivity equivalent after the solution with EIDORS

is implemented, we will have a vector solution of 912 elements. In order to display them on a screen with a color scale, it is necessary to use the library OpenCV. A total of 856 quadrilateral pixels in a 32×32 size screen are displayed. Some of the pixels are not considered for displaying because of the imaging of a circular area and placing white color instead of a color of the conductivity value. In the first implementation, it was used only a gray scale for imaging reconstructed by backprojection. Fig. 3 shows a reconstructed image using the same values computed in the forward problem obtained by EIDORS, it can be noticed that this is the same image since the results of conductivity in each element agree totally. Up to this point the OpenCV library was only used to draw the pixels of the backprojection reconstruction, using drawing functions as *cvRectangle();* for filled pixels with a RGB conductivity value and for the outline of each pixel. Later the OpenCV library was used to improve the presentation on the deployment of the image, adding color to the interface with some features such as the application of a colormap, drawing of a colormap value bar, filtering and generation of multiple images at the same time on the screen.

To validate the obtained results reconstructing an image, a sample was taken and the images were reconstructed in both EIDORS and the developed program. This is an example that contains an object near the center in a circular region. Fig. 4 shows the reconstruction results. Qualitatively it is possible to observe that it is the same object, it must be noticed that since the image is in a gray

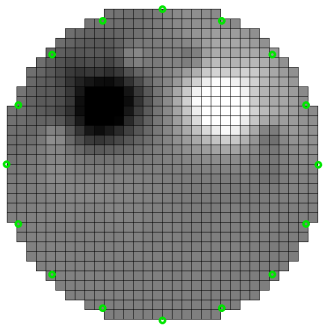


Fig. 3: Backprojection using an interface with C language and a gray scale

scale, it is not possible to make a hard comparison and is required to perform the same reconstruction using colors to highlight contrast and values of specific conductivity in certain regions. The results using colors are shown later in this work. To check the operation of reconstruction with real measurements, some

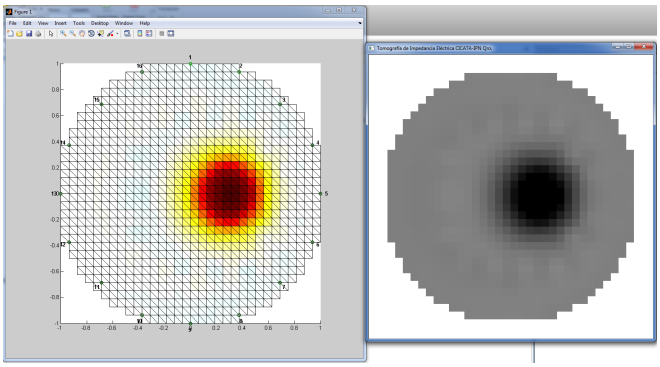


Fig. 4: Comparison of the image output using EIDORS and using the developed interface

real measured data were taken from a test phantom and made at the University of Ottawa in October 2005 [14]. The data were taken in an experiment of a 30cm diameter circular tank with a saline solution of 0.9% and a non-conductive spherical object inside, then the same experiment was performed with two non-conductive spherical objects. The non-conductive object is a golf ball 2cm radio.

In Fig. 5 are shown both reconstructions with the program, in the first one, an object is located at (-7,0) cm (*left*) and in the second one are shown two

objects, one at (0,7) cm and other at (-7,0) cm (*right*). In Fig. 5 you can notice a

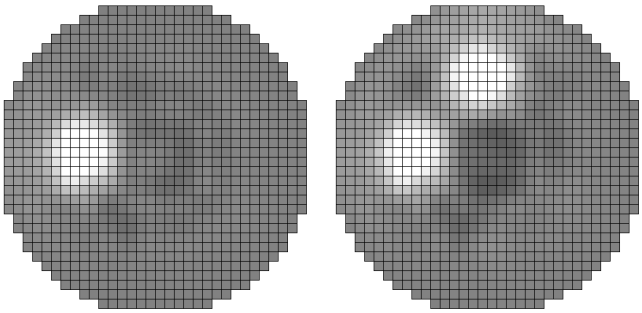


Fig. 5: Real image reconstruction with measurements made in a phantom, *left:* a Golf ball, *right:* two Golf balls

low resolution in the generated image but a high contrast. That is a distribution of the conductivity of a saline solution versus the conductivity of the object, in this case non-conductive; the high contrast highlights shows non-conductive regions like the golf ball clearly, it makes this technique very useful in detecting possible applications as *in vivo* bone density applications.

Once having these results, we proceed to improve the interface, a stage was added into the program that will add color to the grading scale conductivity values. It is a color graduation equivalent to type *jet* colormap. Fig. 6 shows the backprojection reconstruction of the first example with a color scale. In this case the pixel grid appears defining the size of each pixel.

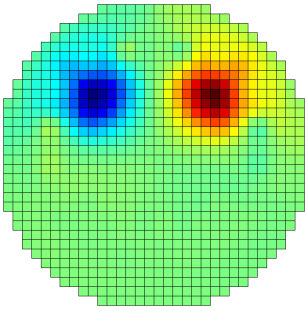


Fig. 6: Backprojection using the equivalent colormap *jet*

The color assignment is performed directly in the program by calculating some numerical values, working in color levels (R, G, B). For gray scale, the allocation is as follows:

```
cc=0.5-data[ ]/2;
r=cc*255;
g=cc*255;
b=cc*255;
```

with *cc* a *double* type variable and r, g, b *integer* type variables. *cc* must be a vector with values in 0 and 255. For the color scale, the next assignment must be used:

```
cc=2*(0.75 - abs(data[ ]));
if(cc>1) cc=1;
if(cc<0) cc=0;
g=cc*255;
cc=1.5 - 2*abs(data[ ] + 0.5);
if(cc>1) cc=1;
if(cc<0) cc=0;
r=cc*255;
cc=1.5 - 2*abs(data[ ] - 0.5);
if(cc>1) cc=1;
if(cc<0) cc=0;
b=cc*255;
```

This for the case of *jet* assignation, where *abs* denotes the absolute value of *data[]-0.5*. For any other color scale, simply must be programmed an assignation based in function of *data[]*, that is the vector containing the values of conductivity after reconstruction.

It is worth mentioning that the interface has the ability to read 208 values of difference potential from the electrode pairs placed on the boundary of the region that is measured. Thus the acquisition of potential is a vector of 16×13 values, since the system was designed for a connection of 16 electrodes, thus a potential vector of readings on the boundary is defined

$$m_{rec} = \left\{ \begin{array}{c} [v_1^1, v_2^1, \cdots, v_{13}^1]^T \\ [v_1^2, v_2^2, \cdots, v_{13}^2]^T \\ \vdots \\ [v_1^{16}, v_2^{16}, \cdots, v_{13}^{16}]^T \end{array} \right\} \tag{6}$$

where v_i^j is the i-th measurement in the j-th injection. With this information is possible to solve quickly the equation (5) for σ_e.

An improvement that can be done is the implementation of a Gaussian filter, using the *GaussianBlur()* function which is included in the openCV library. Filter

parameters applied were the standard deviation kernel address $x = 51$ and the standard deviation $y = 51$.

Fig. 7 shows the interface with *jet* colors and the Gaussian filter applied. The smoothing is chosen arbitrarily in this case, but with lower values than here proposed, results do not distort the image. For this filter are always required

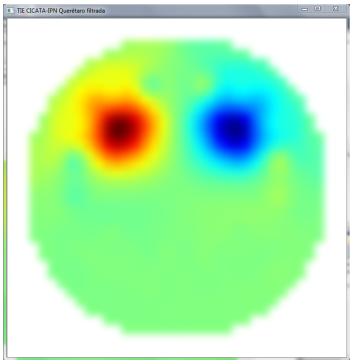


Fig. 7: EIT reconstructed image and filtered using OpenCV

the standard deviation kernel parameters as odd or possibly zero, with the instruction to C ++ language:

```
cv::GaussianBlur(<input array>, <output array>,
kernel size(51,51), border type=0);
```

It is possible to use this information for further analysis of conductivity in regions that show contrast, for example, an application in the future of this work, is the density detection by image, also can be used for forms detection, seeking malformations or for application in other involving Electrical Impedance Tomography.

4 Conclusions

The Sheffield backprojection method for imaging EIT was implemented successfully without using EIDORS and Matlab, instead of this, were used C language and the OpenCV library with the possibility to apply a Gaussian filter and the jet colormap; the image was reconstructed into a level of conductivity values represented by colors, thus we have a friendly graphical interface that displays an EIT image. The reconstruction was made from 208 potential measured values using Sheffield backprojection. Some reconstructions of real data were made to validate and display a color filtered image with contrast.

Acknowledgments The authors would like to express their gratitude to Instituto Politécnico Nacional for the support given in SIP 20151350 and SIP 20161290 projects. Thanks to CONACYT for scholarship (CVU/Scholarship holder) 350051/237150.

References

1. Molinari, M.: High fidelity imaging in electrical impedance tomography, Ph.D. dissertation, University of Southampton, (2003).
2. Packham, M., Barnes, G., Sato dos Santos, G., Aristovich, K., Gilad, O., Ghosh, A., Oh, T. and Holder, D.: Empirical validation of statistical parametric mapping for group imaging of fast neural activity using electrical impedance tomography. Physiological Measurement, vol. 37, no. 6, pp.951-967, (2016).
3. Graham, B.: Enhancements in electrical impedance tomography (eit) image reconstruction for 3d lung imaging, Ph.D. dissertation, University of Ottawa, (2007).
4. Adler, A.: Measurement of pulmonary function with electrical impedance tomography, Ph.D. dissertation, UNIVERSITÉ DE MONTRÉAL, (1995).
5. Adler, A, Arnold, J., Brown, B., Dixon, P., Faes, T., Frerichs, I., Gagnon, H., Garber, Y., Grychtol, B., Hahn, G., Lionheart, W., Malik, A., Patterson, R., Stocks, J., Tizzard, A., Weiler, N. and Wolf, G.: GREIT: a unified approach to 2d linear eit reconstruction of lung images, Physiol. Meas., vol. 30, (2009).
6. Grychtol, B., Mller, B., Adler, A.: 3D EIT image reconstruction with GREIT. Physiol. Meas., vol. 37, no. 6, pp. 785-800, (2016).
7. Lionheart, W.: EIT reconstruction algotithms: pitfalls, challenges and recent developments, Physiol. Meas., (2004).
8. Holder, D.: Electrical Impedance Tomography: Methods, History and Applications. Institute of Physics Publishing, (2005).
9. Antink, C.H., Pikkemaat, R., Malmivuo, J. and Leonhardt, S.: A shape-based quality evaluation and reconstruction method for electrical. Physiological Measurement, vol. 36, no. 6, (2015).
10. Vauhkonen, M.: Electrical Impedance Tomography and prior information, Ph.D. dissertation, Kuopio University, 1997.
11. Santosa, F. and Vogelius, M.: A backprojection algorithm for electrical impedance imaging, SIAM J. Appl. Math., vol. 50, pp. 216243, (1990).
12. Barber, D. and Brown, B.: Applied potential tomography, Journal of Physics E: Scientific Instruments, (1984).
13. Adler, A. and Lionheart, W.R.: Uses and abuses of EIDORS: an extensible software base for EIT, Physiological measurement, vol. 27, no. 5, (2006).
14. Gómez-Laberge, C.: Electrical impedance tomography for deformable media, Master's thesis, University of Ottawa, (2006).

A new associative classification approach to Parkinson pre diagnosis

Rogelio Ramírez-Rubio, Sonia L. Valencia-Ortíz, Mario Aldape-Pérez, Cornelio Yáñez-Márquez, and Oscar Camacho-Nieto

Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC)
Instituto Politécnico Nacional (IPN)
Ciudad de México, México

http://www.aldape.mx

Abstract. In this paper Parkinson Disease pre diagnosis is addressed from a different perspective using associative models. Associative memories are mathematical models used to recognize and classify instances. In the present paper we address Parkinson classification problem using an autoassociative memory and the smallest normalized difference criteria. Parkinson Disease symptoms begin slowly, typically on one side of the body and then both sides are affected. It is important to diagnose this disorder as soon as possible in order to reduce its consequences. In this paper, a set of well-known classifiers are compared in order to have a fair classification performance comparison.

Keywords: Associative models, CHAT, Parkinson Disease, Pattern Recognition.

1 Introduction

Associative models have been used mainly to perform pattern recognition, but they are also useful to perform classification tasks. It has to be mentioned that associative memories in its autoassociative mode have not been widely used in classification tasks. In this paper we propose an algorithm that uses an autoassociative memory as its first step and smallest normalized difference criteria as its second step. Parkinson Disease has been addressed with many classification algorithms and diagnosis tests [1,2,3].

Parkinson Disease is a type of movement disorder. It occurs when nerve cells (neurons) do not produce enough of an important chemical in the brain called dopamine. Some cases are genetic, but most do not seem to occur between members of the same family. Symptoms begin slowly, in general, on one side of the body. Then affect both sides [4]. Some symptoms are:

– Trembling hands, arms, legs, jaw and face

– Stiffness in the arms, legs and trunk
– Slowness of movement
– Balance and coordination problems

Parkinson Disease is a neurological disorder with evolving layers of complexity. It has long been characterized by the classical motor features of Parkinsonism associated with Lewy bodies and loss of dopaminergic neurons in the substantia nigra. However, the symptomatology of Parkinson's disease is now recognized as heterogeneous, with clinically significant non-motor features [5].

Normal maintenance of human motivation depends on the integrity of subcortical structures that link the prefrontal cortex with the limbic system. Structural and functional disruption of different networks within these circuits alters the maintenance of spontaneous mental activity and the capacity of affected individuals to associate emotions with complex stimuli. The clinical manifestations of these changes include a continuum of abnormalities in goal-oriented behaviors known as apathy. Apathy is highly prevalent in Parkinson's disease (and across many neurodegenerative disorders) and can severely affect the quality of life of both patients and caregivers. Differentiation of apathy from depression, and discrimination of its cognitive, emotional, and auto-activation components could guide an individualized approach to the treatment of symptoms [6].

The paper is organized as follows. A succinct description of associative memories fundamentals is presented in Section 2. Section 3 provides a concise description of the most important characteristics of Alpha-Beta Associative Memories. In Section 4 our proposal foundations are presented. In Section 5 classification accuracy results achieved by each one of the compared algorithms using Parkinson Disease dataset are presented. Finally, our proposal advantages, as well as some conclusions will be discussed in section 7.

2 Associative Memories

An associative memory $\mathbf{M}$ is a mathematical model that relates input patterns and output patterns. Each input vector $\mathbf{x}$ forms an association with its corresponding output vector $\mathbf{y}$. For each γ integer and positive, the corresponding association will be denoted as: $(\mathbf{x}^\gamma, \mathbf{y}^\gamma)$. An associative memory $\mathbf{M}$ is represented by a matrix whose ij-th component is m_{ij}. An associative memory $\mathbf{M}$ is generated from an *a priori* finite set of known associations, called the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $\mathbf{x}^\mu = \mathbf{y}^\mu \; \forall \mu \in \{1, 2, ..., p\}$, $\mathbf{M}$ is autoassociative, otherwise it is heteroassociative; in this case, it is possible to establish that $\exists \mu \in \{1, 2, ..., p\}$ for which $\mathbf{x}^\mu \neq \mathbf{y}^\mu$. If we consider the fundamental set of patterns $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$ where n and m are the dimensions of input patterns and output patterns, respectively,

it is said that $\mathbf{x}^\mu \in A^n$, $A = \{0, 1\}$ and $\mathbf{y}^\mu \in A^m$. Then the j-th component of an input pattern $\mathbf{x}^\mu$ is $x_j^\mu \in A$. Analogously, the i-th component of an output pattern $\mathbf{y}^\mu$ is represented as $y_i^\mu \in A$. Therefore, the fundamental input patterns and output patterns are represented as follows:

$$\mathbf{x}^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \qquad \mathbf{y}^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

A distorted version of a pattern $\mathbf{x}^\gamma$ to be recalled will be denoted as $\widetilde{\mathbf{x}}^\gamma$. An unknown input pattern to be recalled will be denoted as $\mathbf{x}^\omega$. If when an unknown input pattern $\mathbf{x}^\omega$ is fed to an associative memory $\mathbf{M}$, happens that the output corresponds exactly to the associated pattern $\mathbf{y}^\omega$, it is said that recalling is correct.

3 Alpha-Beta Associative Memories

Alpha-Beta Associative Memories were first introduced in [7]. Alpha-Beta Associative Memories mathematical foundations are based on two binary operators: α and β. Alpha operator is used during the learning phase, while Beta operator is used during the recalling phase. The mathematical properties within these operators, allow the $\alpha\beta$ associative memories to exhibit similar characteristics to the binary version of the morphological associative memories, in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall [8]. First, we define set $A = \{0, 1\}$ and set $B = \{0, 1, 2\}$, so α operator is defined as in Table 1 and β operator is defined as in Table 2. These two binary operators along with maximum ($\vee$) and minimum ($\wedge$) operators establish the mathematical tools around the Alpha-Beta model [9]. The definitions of α and β exposed in Table 1 and Table 2 , imply that: α is increasing by the left and decreasing by the right, β is increasing by the left and right, β is the left inverse of α, see Table 3. A summary of the mathematical properties of α and β operators are shown in Table 4 and Table 5.

According to the operator that is used during the learning phase, two kinds of Alpha-Beta Associative Memories are obtained. If maximum operator ($\vee$) is used, Alpha-Beta Associative Memory MAX type will be obtained, denoted as $\mathbf{M}$; analogously, if minimum operator ($\wedge$) is used, Alpha-Beta Associative Memory min type will be obtained, denoted as $\mathbf{W}$ [7].

In order to understand how the learning and recalling phases are carried out, some matrix operations definitions are required.

Table 1: Alpha Operator.
$\alpha : A \times A \longrightarrow B$

x	y	α(x,y)
0	0	1
0	1	0
1	0	2
1	1	1

Table 2: Beta Operator.
$\beta : B \times A \longrightarrow A$

x	y	β(x,y)
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Definition 1. *Let n and m be integer and positive numbers that represent the dimension of input patterns and output patterns, respectively. α max Operation is defined according to the following expression:*

$$P_{mxr} \nabla_{\alpha} Q_{rxn} = \left[f_{ij}^{\alpha} \right]_{mxn} \tag{1}$$

where $f_{ij}^{\alpha} = \vee_{k=1}^{r} \ \alpha(p_{ik}, q_{kj})$

Definition 2. *Let n and m be integer and positive numbers that represent the dimension of input patterns and output patterns, respectively. β max Operation is defined according to the following expression:*

$$P_{mxr} \nabla_{\beta} Q_{rxn} = \left[f_{ij}^{\beta} \right]_{mxn} \tag{2}$$

where $f_{ij}^{\beta} = \vee_{k=1}^{r} \ \beta(p_{ik}, q_{kj})$

Definition 3. *Let n and m be integer and positive numbers that represent the dimension of input patterns and output patterns, respectively. α min Operation is defined according to the following expression:*

$$P_{mxr} \Delta_{\alpha} Q_{rxn} = \left[f_{ij}^{\alpha} \right]_{mxn} \tag{3}$$

where $f_{ij}^{\alpha} = \wedge_{k=1}^{r} \ \alpha(p_{ik}, q_{kj})$

Table 3: Operators Properties.

$\beta[\alpha(x,y),y]=x$
$\beta[\alpha(x,y),x]=x$
$\beta[\alpha(x,x),y]=y$

Table 4: Alpha Operator Properties.

$\alpha : A \times A \longrightarrow B$

$\alpha(x,x)=1$
$(x \leq y) \longleftrightarrow [\alpha(x,y) \leq \alpha(y,x)]$
$(x \leq y) \longleftrightarrow [\alpha(x,z) \leq \alpha(y,z)]$
$(x \leq y) \longleftrightarrow [\alpha(z,x) \geq \alpha(z,y)]$

Definition 4. *Let n and m be integer and positive numbers that represent the dimension of input patterns and output patterns, respectively. $\beta \max$ Operation is defined according to the following expression:*

$$P_{mxr} \Delta_\beta Q_{rxn} = \left[f_{ij}^{\beta} \right]_{mxn} \tag{4}$$

where $f_{ij}^{\beta} = \wedge_{k=1}^{r} \ \beta(p_{ik}, q_{kj})$

Whenever a column vector of dimension m is operated with a row vector of dimension n, both operations ∇_α and Δ_α, are represented by $\oplus$; consequently, the following expression is valid:

$$\mathbf{y} \boldsymbol{\nabla}_{\boldsymbol{\alpha}} \mathbf{x}^t = \mathbf{y} \oplus \mathbf{x}^t = \mathbf{y} \boldsymbol{\Delta}_{\boldsymbol{\alpha}} \mathbf{x}^t \tag{5}$$

If we consider the fundamental set of patterns $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$ then the ij-th entry of the matrix $\mathbf{y}^\mu \oplus (\mathbf{x}^\mu)^t$ is expressed as follows:

$$\left[\mathbf{y}^\mu \oplus (\mathbf{x}^\mu)^t \right]_{ij} = \alpha(y_i^\mu, x_j^\mu) \tag{6}$$

3.1 Learning Phase

Find the adequate operators and a way to generate a matrix $\mathbf{M}$ that will store the p associations of the fundamental set $\left\{ \left(\mathbf{x}^1, \mathbf{y}^1\right), \left(\mathbf{x}^2, \mathbf{y}^2\right), ..., \left(\mathbf{x}^p, \mathbf{y}^p\right) \right\}$, where $\mathbf{x}^\mu \in A^n$ and $\mathbf{y}^\mu \in A^m \ \forall \mu \in \{1, 2, ..., p\}$.

Step 1. For each fundamental pattern association $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$, generate p matrices according to the following rule:

$$\left[\mathbf{y}^\mu \oplus (\mathbf{x}^\mu)^t \right]_{mxn} \tag{7}$$

Table 5: Beta Operator Properties.

$$\beta : B \times A \longrightarrow A$$

$\beta(1, x) = x$
$\beta(x, x) = x, \forall x \in A$
$(x \leq y) \rightarrow [\beta(x, z) \leq \beta(y, z)]$
$(x \leq y) \rightarrow [\beta(z, x) \leq \beta(z, y)]$

Step 2. In order to obtain an Alpha-Beta Associative Memory MAX type, apply the binary MAX operator ($\vee$) according to the following rule:

$$\mathbf{M} = \vee_{\mu=1}^{p} \left[\mathbf{y}^{\mu} \oplus (\mathbf{x}^{\mu})^{t} \right] \tag{8}$$

Step 3. In order to obtain an Alpha-Beta Associative Memory min type, apply the binary min operator ($\wedge$) according to the following rule:

$$\mathbf{W} = \wedge_{\mu=1}^{p} \left[\mathbf{y}^{\mu} \oplus (\mathbf{x}^{\mu})^{t} \right] \tag{9}$$

Consequently, the ij-th entry of an Alpha-Beta Associative Memory MAX type is given by the following expression:

$$\nu_{ij} = \vee_{\mu=1}^{p} \alpha(y_i^{\mu}, x_j^{\mu}) \tag{10}$$

Analogously, the ij-th entry of an Alpha-Beta Associative Memory min type is given by the following expression:

$$\psi_{ij} = \wedge_{\mu=1}^{p} \alpha(y_i^{\mu}, x_j^{\mu}) \tag{11}$$

3.2 Recalling Phase

Find the adequate operators and sufficient conditions to obtain the fundamental output pattern $\mathbf{y}^{\mu}$, when either memory $\mathbf{M}$ or memory $\mathbf{W}$ is operated with the fundamental input pattern $\mathbf{x}^{\mu}$.

Step 1. A pattern $\mathbf{x}^{\omega}$, with $\omega \in \{1, 2, ..., p\}$, is presented to the Alpha-Beta Associative Memory, so $\mathbf{x}^{\omega}$ is recalled according to one of the following rules.

Alpha-Beta Associative Memory MAX type:

$$\mathbf{M} \Delta_{\beta} \mathbf{x}^{\omega} = \wedge_{j=1}^{n} \beta(\nu_{ij}, x_j^{\omega}) = \wedge_{j=1}^{n} \left\{ \left[\vee_{\mu=1}^{p} \alpha(y_i^{\mu}, x_j^{\mu}) \right], x_j^{\omega} \right\}$$

Alpha-Beta Associative Memory min type:

$$\mathbf{W} \bigtriangledown_{\beta} \mathbf{x}^{\omega} = \vee_{j=1}^{n} \beta(\psi_{ij}, x_j^{\omega}) = \vee_{j=1}^{n} \left\{ \left[\wedge_{\mu=1}^{p} \alpha(y_i^{\mu}, x_j^{\mu}) \right], x_j^{\omega} \right\}$$

Without dependence on the Alpha-Beta Associative Memory type used throughout the recalling phase, a column vector of dimension n will be obtained.

Remarks

- Advantages
  One of the biggest advantage of Alpha-Beta Associative Memories is that this mathematical model recalls the fundamental set completely, if it is trained in auto-associative mode. This implies that all the fundamental patterns that are used along the learning phase will be retrieved without errors. The proof of the theorem that guarantees the complete recovery of the fundamental set can be found in [7].
- Disadvantages
  The main disadvantage with this mathematical model is that it only works with binary patterns, so in case you want to work with patterns with real components, each component has to be binary coded. As a consequence, data processing complexity is increased. It should be noted that this mathematical model is very robust to additive or subtractive noise in input patterns [7], however, this model has low classification performance when you have mixed noise in input patterns [7]. These associative memories have been widely used in many applications but the fundamental input patterns had to be coded using the Johnson-Möbius Modified Code [7]. This type of coding mantains the order relation between patterns. As a result noise type is preserved and performance is improved, but also processing complexity is increased.

4 SND Associative Memory

In this section, Smallest Normalized Difference Associative Memory (SNDAM) theoretical foundations are presented. In order to eliminate Alpha-Beta Associative Memories disadvantages, we have to extend Alpha and Beta operators to $\mathbb{R}$ domain. Alpha operation has only one case of application; however, Beta operation has two cases according to the type of memory that is built in the training phase (MAX or min).

Definition 5. *Alpha operation* $\alpha_{\mathbb{R}} : \mathbb{R}\ x\ \mathbb{R} \to \mathbb{R}$ *is defined as follows:*

$$\alpha_{\mathbb{R}}(c, d) = c - d + 1 \tag{12}$$

Definition 6. *Beta* MAX *operation* $\beta^{\vee}_{\mathbb{R}} : \mathbb{R}\ x\ \mathbb{R} \to \mathbb{R}$ *is defined as follows:*

$$\beta^{\vee}_{\mathbb{R}}(c, d) = \begin{cases} d - |c| - 1 & \text{if } c \neq d \\ c & \text{if } c = d \end{cases} \tag{13}$$

Definition 7. *Beta min operation* $\beta^{\wedge}_{\mathbb{R}} : \mathbb{R}\ x\ \mathbb{R} \to \mathbb{R}$ *is defined as follows:*

$$\beta^{\wedge}_{\mathbb{R}}(c, d) = \begin{cases} c - |d| - 1 & \text{if } c \neq d \\ c & \text{if } c = d \end{cases} \tag{14}$$

It is important to note that when an Alpha-Beta Associative Memory is trained in auto-associative mode, the main diagonal of the learning matrix has only 1's. As a consequence, complete recalling of the fundamental set is guaranteed (see Equation 15). The proof of the theorem that guarantees that an Alpha-Beta Associative Memory recovers completely the training set, appears in [7].

$$\alpha_{\mathbb{R}}(c, c) = c - c + 1 = 1, \forall c \in \mathbb{R} \tag{15}$$

4.1 Algorithm

Once we have extended Alpha and Beta operators for real values, we can describe the algorithm in both of its phases: training and recalling. The proposed algorithm consists of two stages. In the first stage, an associative memory is built, while in the second, the smallest normalized distance is applied to the recovered pattern in order to assign a class label.

Definition 8. *Let p be the cardinality of the fundamental set of associations and let n be the dimension of fundamental input patterns $\mathbf{x}^{\mu}$. Let $\mathbf{x}^{MAX}$ be the vector that stores the maximum value $\forall j \in \{1, 2, ..., n\}$, according to the following expression:*

$$x_j^{MAX} = \bigvee_{\mu=1}^{p} x_j^{\mu} \tag{16}$$

Training phase Find adequate operators and a way to generate an associative memory $\mathbf{M}$ that will store p associations of the fundamental set. Let p be the cardinality of the fundamental set of associations and let n be the dimension of fundamental input patterns, whose n components are in $\mathbb{R}$ domain.

1. In order to build an Alpha-Beta Associative Memory MAX type, use Equation 17, similarly, if you want to build an Alpha-Beta Associative Memory min type, use Equation 18.

$$\mathbf{M} = \bigvee_{\mu=1}^{p} \left(\alpha_{\mathbb{R}} \left(\mathbf{x}^{\mu}, \mathbf{x}^{\mu}\right)\right) \tag{17}$$

$$\mathbf{M} = \bigwedge_{\mu=1}^{p} \left(\alpha_{\mathbb{R}} \left(\mathbf{x}^{\mu}, \mathbf{x}^{\mu}\right)\right) \tag{18}$$

   Where $\bigvee$ is the maximum operator and $\bigwedge$ is the minimum operator. After this step, we get an associative memory $\mathbf{M}$.
2. Search for the highest absolute value of each component, using the maximum vector $\mathbf{x}^{MAX}$ as stated in Equation 16.

Recalling phase After we have trained our associative memory and found the $\mathbf{x}^{MAX}$ vector, execute the following steps:

1. Recall pattern $\mathbf{y}$ from an unknown input pattern $\tilde{\mathbf{x}}$ using Equation 19 or Equation 20:
$$\mathbf{y} = \bigwedge (\beta(\mathbf{M}, \tilde{\mathbf{x}})) \tag{19}$$
$$\mathbf{y} = \bigvee (\beta(\mathbf{M}, \tilde{\mathbf{x}})) \tag{20}$$
Where $\bigvee$ is the maximum operator and $\bigwedge$ is the minimum operator
2. Compute the normalized difference $\boldsymbol{\delta}^{\mu}$ between the recalled pattern $\mathbf{y}$ and the fundamental input patterns $\mathbf{x}^{\mu}$, $\forall \mu \in \{1, 2, ..., p\}$ as stated in Equation 21
$$\boldsymbol{\delta}^{\mu} = \sum_{i=1}^{n} \frac{|y_i - x_i^{\mu}|}{x_i^{max}} \tag{21}$$
3. Obtain the smallest normalized difference value $\boldsymbol{\delta}^{\mu}$ in order to identify μ
4. Use μ value to assign the class label of the pattern $\mathbf{x}^{\mu}$ to the recalled pattern $\mathbf{y}$.

5 Experimental Phase

Throughout the experimental phase, Parkinson Disease Dataset was used as test set to estimate the classification performance of each one of the compared algorithms. These dataset was taken from the UCI machine learning repository [10], from which full documentation can be obtained. SNDAM performance was compared against the performance achieved by the twenty best-performing algorithms of the seventy-six available in WEKA 3: Data Mining Software in Java [11].

5.1 Parkinson Dataset

This database was created by Max Little of the University of Oxford in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the dataset is a particular voice measure, and each row corresponds to one of 195 voice recording from these individuals ("name" column). The main purpose of this dataset is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

Table 6: Classification accuracy was computed using Stratified 10-Fold Cross Validation.

	Algorithm	% Accuracy
1.	AdaBoostM1	85.12
2.	Bagging	87.69
3.	BayesNet	80.00
4.	Dagging	85.12
5.	DecisionTable	83.58
6.	DTNB	85.12
7.	FT	84.61
8.	LMT	86.15
9.	Logistic	86.66
10.	MultiClassClassifier	86.66
11.	NaiveBayes	69.23
12.	NaiveBayesSimple	69.23
13.	NveBayesUpdateable	69.23
14.	RandomCommittee	90.76
15.	RandomForest	90.76
16.	RandomSubSpace	88.7
17.	RBFNetwork	84.10
18.	RotationForest	90.25
19.	SimpleLogistic	84.61
20.	SMO	87.17
★	SNDAM	92.22

6 Results

The experimental phase was conducted with twenty algorithms. All of them are different pattern classification methods, executed in WEKA environment [11]. The algorithms accuracy is shown in Table 6. The cross validation method was the Stratified 10-Fold Cross Validation [12] in order to do an objective comparison.

The first twenty algorithms were tested because all of them represent multiple ways of classification and are widely used in pattern recognition tasks. It is worth to say there are other classification algorithms and previous work in associative classifications algorithms as can be seen in [13,14,15], notwithstanding our proposal achieve the highest classification accuracy, specifically, it has an advantage of 22.99% in comparison with NaiveBayes, NaiveBayesSimple, and NveBayesUpdateable (whose results were the worst in our test), likewise, it has an advantage of 1.46% in comparison with RandomCommittee and RandomForest (whose results are the best in our test, except for our proposal).

7 Conclusion

We can see that there are algorithms that cannot classify competitively (like NaiveBayes, NaiveBayesSimple, NveBayesUpdateable) because of their design. Also we can see that there are algorithms that can classify competitively and even more (like RandomCommittee, and RandomForest), but it's worth to say that our proposal has achieved the best performance in this Parkinson Disease classification, with an accuracy of 92.22%. We can say then that our proposal could be an interesting way to implement Parkinson pre diagnosis.

Acknowledgments The authors of the present paper would like to thank the following institutions for their economical support to develop this work: Science and Technology National Council of Mexico (CONACyT), SNI, National Polytechnic Institute of Mexico (COFAA, SIP, CIDETEC, and CIC).

References

1. Naranjo, L., Pérez, C.J., Campos-Roca, Y., Martín, J.: Addressing voice recording replications for parkinsonŠs disease detection. Expert Systems with Applications **46** (2016) 286 – 292
2. Rana, B., Juneja, A., Saxena, M., Gudwani, S., Kumaran, S.S., Agrawal, R., Behari, M.: Regions-of-interest based automated diagnosis of parkinsonŠs disease using t1-weighted {MRI}. Expert Systems with Applications **42**(9) (2015) 4506 – 4516
3. Castelli, M., Vanneschi, L., Silva, S.: Prediction of the unified parkinsonŠs disease rating scale assessment using a genetic programming system with geometric semantic genetic operators. Expert Systems with Applications **41**(10) (2014) 4608 – 4616
4. Kohonen, T.: Correlation matrix memories. IEEE Transactions on Computers **C-21**(4) (1972) 353–359
5. Lee, A., Gilbert, R.M.: Epidemiology of parkinson disease. Neurologic Clinics **34**(4) (2016) 955 – 965
6. Tokuchi, R., Hishikawa, N., Sato, K., Hatanaka, N., Fukui, Y., Takemoto, M., Ohta, Y., Yamashita, T., Abe, K.: Differences between the behavioral and psychological symptoms of alzheimer's disease and parkinson's disease. Journal of the Neurological Sciences **369** (2016) 278 – 282
7. Yánez-Márquez, C.: Associative Memories based on Order Relations and Binary Operators (In Spanish). PhD thesis, Instituto Politécnico Nacional (2002)
8. Ritter, G.X., Urcid, G., Iancu, L.: Reconstruction of patterns from noisy inputs using morphological associative memories. Journal of Mathematical Imaging and Vision **19**(2) (2003) 95–111
9. Yáñez-Márquez, C., Díaz-de-León., J.L.: Memorias asociativas basadas en relaciones de orden y operaciones binarias. Computación y Sistemas **6**(4) (2003)
10. Asuncion, A., Newman, D.: UCI Machine Learning Repository (2007)

11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations **11**(1) (2009) 10–18
12. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97**(1) (1997) 273–324
13. Aldape-Pérez, M., Yánez-Márquez, C., Camacho-Nieto, O., Argüelles-Cruz, A.J.: An associative memory approach to medical decision support systems. Computer Methods and Programs in Biomedicine **106** (May 2012) 287–307
14. López-Yánez, I., Yánez-Márquez, C., Camacho-Nieto, O., Aldape-Pérez, M., Argüelles-Cruz, A.J.: Collaborative learning in postgraduate level courses. Computers in Human Behavior **51** (2015) 938–944
15. Aldape-Pérez, M., Yánez-Márquez, C., Camacho-Nieto, O., López-Yánez, I., Argüelles-Cruz, A.J.: Collaborative learning based on associative models: Application to pattern classification in medical datasets. Computers in Human Behavior **51** (2015) 771–779

Dominant Genetic Algorithm for Feature Selection with Associative Models

José A. Estrada-Pavía, Mario Aldape-Pérez, and Oscar Camacho-Nieto

Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC)
Instituto Politécnico Nacional (IPN)
Ciudad de México, México
jestradap@ipn.mx; maldape@ipn.mx; oscarc@cic.ipn.mx;
http://www.aldape.mx

Abstract. In machine learning and statistics, feature selection, is the process of selecting a subset of relevant features that helps to decrease the dimension of a set of patterns and to improve classification performance. Feature Selection aims to obtain the least subset of features that represent the problem In this paper a novel algorithm is presented. It is called Dominant Genetic Algorithm (DGA). This genetic algorithm is a wrapper style feature selection algorithm and the search strategy is guided by an hybrid associative classifier. The classifier evaluates different solutions and gives them a value based on classification accuracy. Obtained results suggest that the algorithm can be used to improve classification performance. Experimental phase was performed using five data sets, widely used in machine learning and statistics.

Keywords: Associative models, CHAT, DGA, Genetic Algorithms, Pattern Recognition.

1 Introduction

Every day more data is generated, but all that data has no meaning until a process transform it into information. One way to obtain information from data is by classifying them. When the data is classified, it is desirable to have as less characteristics as possible, this means that is desirable to perfom a feature selection process before classifying.

The best way to take on feature selection is with an exhaustive search of the best subset of characteristics, this can be done only in datasets where there are not many characteristics. In those datasets with huge amount of features, others methods, instead of exhaustive searching must be looked for [1].

Feature selection algorithms may roughly be divided into three types: The first type is the set of algorithms that are build into adaptive systems for data analysis, called predictors. The second type are wrapper around predictors,

providing them a subset of features and receiving their feedback, those algorithms are known as wrapper algorithms. The third type are independent algorithms from predictors, these algorithms filter the data that have little chance to be useful in classification, these filter methods are based on performance evaluation metric, calculated directly from the data [2].

Feature selection can be analyzed as a search process, where the search must be guided by the classification performance and also by the number of selected characteristics. Search processes are optimization problems that can be solved using genetic algorithms. The use of a genetic algorithm to solve the problem of feature selection was introduced by Siedlecki and Sklansky[3]. Also Ludmila Kuncheva used a genetic algorithm in feature selection for parallel classifiers [4,5].

In this paper, a genetic algorithm wrapper-based method for feature selection is proposed, the DGA. This method employs an associative classifier as the method for evaluating feature subsets. The use of a wrapper-style feature selection method with an associative classifier was first proposed in [6]. This proposal executes an exhaustive search using parallel processing to improve the classification performance. However, that approach is still too time-consuming. This work aims to analyze a suitable algorithm to improve the feature selection in big data sets. The rest of the paper is organized as follows. Section 2 provides a brief overview of genetic algorithms. Section 4 explains the associative classifier employed in the proposed model. The proposed model is described in Section 5 and the experimental results are presented in Section 6. To conclude, Section 7 presents a discussion on the obtained results.

2 Genetic Algorithms

Genetic Algorithms (GA) are search procedures based on natural selection and natural genetics. The first GA was developed by John H. Holland in 1960 to allow computers to evolve solutions to difficult search and combinatorial problems, such as function optimization and machine learning [7]. Also the genetic algorithm can be defined as a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects, each with an associated fitness value, into a new population using operations patterned after the Darwinian principle of reproduction and survival of the fittest and after naturally occurring genetic operations (notably sexual recombination) [8].

Recently GA have been applied in different areas like in transit network design problem [9], projects of public illumination [10] and in text classification [11], among others.

It is said that most of GA methods have at least four elements in common: Population of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring. The chromosomes are commonly represented as a binary array with n alleles that can get the values

0 or 1. Each chromosome can be thought of as a point in the search space of candidate solutions. The GA processes populations of chromosomes, successively replacing one population with another. The GA requires a fitness function that assigns a score to each chromosome in the current population. The fitness of a chromosome depends on how well that chromosome solves the problem at hand.

The simplest form of genetic algorithm involves three operators: selection, crossover and mutation, each operator is explained below:

Selection: The function of this operation is to select elitist individuals as parents in current population, which can generate offspring. Fitness values are used as criteria to judge whether individuals are elitist [12]**.** The purpose of selection is to emphasize the fitter individuals in the population hoping that their offspring will in turn have even higher fitness [13].

Crossover: This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example the chromosomes 11110000 and 10100011 could be crossed over by the midpoint of each, to produce the two offspring 11110011 and 10100000. The objective of this operator is to recombine building blocks (schemas) on the different solutions.

Mutation: This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string, with some probability, usually very small (e.g., 0.001). Mutation ensures the population against permanent fixation at any particular local result.

According to De Jong [14] a standard GA is:

- Randomly generate a population of m parents.
- Repeat:
  - Compute and save the fitness u(i) for each individual i in the current parent population.
  - Define selection probabilities p(i) for each parent i so that p(i) is proportional to u(i).
  - Generate m offspring by probabilistically selecting parents to produce offspring.
  - Select only the offspring to survive.
- End Repeat

3 Associative Memories

An associative memory $\mathbf{M}$ is a system that relates input patterns and output patterns as follows:

$$\mathbf{x} \rightarrow \boxed{\mathbf{M}} \rightarrow \mathbf{y}$$

with $\mathbf{x}$ and $\mathbf{y}$ the input and output pattern vectors, respectively. Each input vector forms an association with its corresponding output vector. For each γ

integer and positive, the corresponding association will be denoted as: $(\mathbf{x}^\gamma, \mathbf{y}^\gamma)$. An associative memory $\mathbf{M}$ is represented by a matrix whose ij-th component is m_{ij}. An associative memory $\mathbf{M}$ is generated from an *a priori* finite set of known associations, called the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $\mathbf{x}^\mu = \mathbf{y}^\mu \ \forall \mu \in \{1, 2, ..., p\}$, $\mathbf{M}$ is autoassociative, otherwise it is heteroassociative; in this case, it is possible to establish that $\exists \mu \in \{1, 2, ..., p\}$ for which $\mathbf{x}^\mu \neq \mathbf{y}^\mu$. If we consider the fundamental set of patterns $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$ where n and m are the dimensions of the input patterns and output patterns, respectively, it is said that $\mathbf{x}^\mu \in A^n$, $A = \{0, 1\}$ and $\mathbf{y}^\mu \in A^m$. Then the j-th component of an input pattern $\mathbf{x}^\mu$ is $x_j^\mu \in A$. Analogously, the i-th component of an output pattern $\mathbf{y}^\mu$ is represented as $y_i^\mu \in A$. Therefore, the fundamental input and output patterns are represented as follows:

$$\mathbf{x}^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \qquad \mathbf{y}^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

A distorted version of a pattern $\mathbf{x}^\gamma$ to be recalled will be denoted as $\widetilde{\mathbf{x}}^\gamma$. An unknown input pattern to be recalled will be denoted as $\mathbf{x}^\omega$. If when an unknown input pattern $\mathbf{x}^\omega$ is fed to an associative memory $\mathbf{M}$, happens that the output corresponds exactly to the associated pattern $\mathbf{y}^\omega$, it is said that recalling is correct [15].

4 CHAT Associative Memory

CHAT Associative Memory is based on the combination of two associative memories, the Lernmatrix from Steinbuch [16,17] and Linear Associator from Kohonen [18]. The algorithm is as follows:

1. Lets define a fundamental set of input patterns of dimension n with real values on their components, these patterns are organized in m different classes.
2. For each input pattern in the class k define a vector made of zeros except the $k - th$ coordinate, where the value is one.
3. Obtain the medium vector from the fundamental set of patterns according to the following expression:

$$\overline{\mathbf{x}} = \frac{1}{p} \sum_{\mu=1}^{p} \mathbf{x}^\mu \tag{1}$$

With $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, ..., \mathbf{x}^p$ as the set of input patterns
With $\bar{\mathbf{x}}$ as the medium vector for all input patterns
With p as the total number of patterns to be used

4. Take the components from the medium vector as the center of a new set of coordinates.
5. Move all the patterns from the fundamental set according to the following expression:

$$\mathbf{x}^{\mu'} = [\mathbf{x}^{\mu} - \overline{\mathbf{x}}] \ \ \forall \mu \in \{1, 2, ..., p\} \tag{2}$$

With $\mathbf{x}^{\mu'}$ as the displaced pattern
With $\mathbf{x}^{\mu}$ as the original pattern
With $\bar{\mathbf{x}}$ as the medium vector for all input patterns

6. Obtain an associative memory $\mathbf{M}$ by performing the following steps:
   - Given the fundamental set of associations $\{(\mathbf{x}^{\mu}, \mathbf{y}^{\mu}) \mid \mu = 1, 2, ..., p\}$, obtain the displaced fundamental set of associations $\{(\mathbf{x}^{\mu'}, \mathbf{y}^{\mu'}) \mid \mu = 1, 2, ..., p\}$ using expression (1) and expression (2).
   - Consider each one of the p associations $(\mathbf{x}^{\mu'}, \mathbf{y}^{\mu'})$, so an $m \times n$ matrix is obtained according to the following expression:

$$\mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = \begin{pmatrix} y_1^{\mu'} \\ y_2^{\mu'} \\ \vdots \\ y_m^{\mu'} \end{pmatrix} \cdot \left(x_1^{\mu'}, x_2^{\mu'}, ..., x_n^{\mu'}\right) \tag{3}$$

$$\mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = \begin{pmatrix} y_1^{\mu'} x_1^{\mu'} & \cdots & y_1^{\mu'} x_j^{\mu'} & \cdots & y_1^{\mu'} x_n^{\mu'} \\ \vdots & & \vdots & & \vdots \\ y_i^{\mu'} x_1^{\mu'} & \cdots & y_i^{\mu'} x_j^{\mu'} & \cdots & y_i^{\mu'} x_n^{\mu'} \\ \vdots & & \vdots & & \vdots \\ y_m^{\mu'} x_1^{\mu'} & \cdots & y_m^{\mu'} x_j^{\mu'} & \cdots & y_m^{\mu'} x_n^{\mu'} \end{pmatrix}$$

   - Obtain an associative memory $\mathbf{M}$ by adding all the p matrices according to the following expression:

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^{\mu'} \cdot \left(\mathbf{x}^{\mu'}\right)^t = [m_{ij}]_{m \times n} \tag{4}$$

in this way the ij-th component of an associative memory $\mathbf{M}$ is expressed as follows:

$$m_{ij} = \sum_{\mu=1}^{p} y_i^{\mu'} x_j^{\mu'} \tag{5}$$

7. Apply the recalling phase as follows:

The recalling phase pretend to find the class label that belongs to an unknown input vector $\mathbf{x}^{\omega} \in A^n$. Find the class means to obtain the components from the vector $\mathbf{y}^{\omega} \in A^p$ that belongs to the pattern $\mathbf{x}^{\omega}$. The recalled output pattern is obtained according to the following expression:

$$y_i^{\omega} = \begin{cases} 1 \text{ if } \sum_{j=1}^{n} m_{ij} \cdot x_j^{\omega} = \theta \\ 0 \quad \text{otherwise} \end{cases} \tag{6}$$

5 Proposed Method

This paper presents a wrapper-style feature selection algorithm based on a Genetic Algorithm and an associative classifier. The objective of this algorithm is to achieve a solution near to the optimal, exploring as less solution space as possible. The proposed feature selection method is described in Fig. 1.

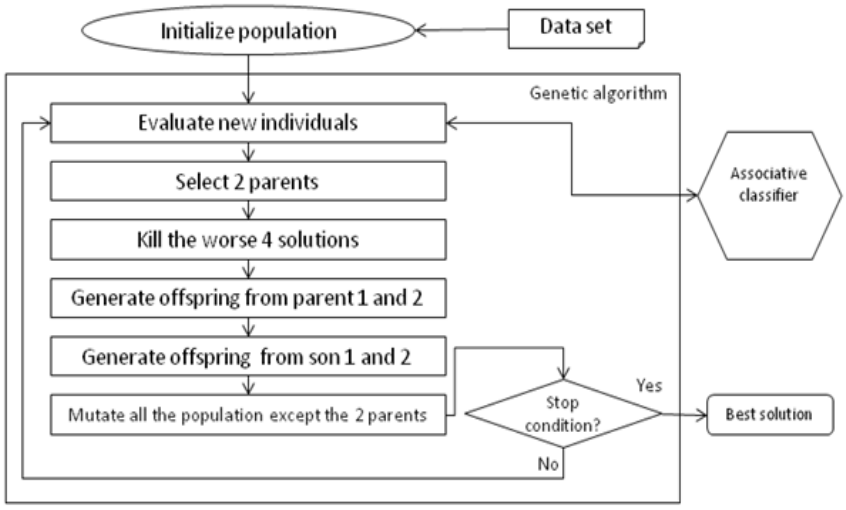


Fig. 1: Proposed method

As can be seen on the Fig. 1 the proposed algorithm is different from conventional GA, first in DGA an initial population of 6 is chosen randomly, this

Table 1: Characteristics of datasets used in the experimental phase.

Data set	No. of features	Solution space
Breast Cancer	9	512
Heart	13	8,192
Hepatitis	19	524,288
Lung Cancer	56	1.44E17
Arrhythmia	279	9.71E83

little population helps to decrease the number of the solutions explored. All the individuals are binary coded, where 1 represents the use of a specific characteristic and 0 represents a characteristic that is not selected. Instead of selecting parents randomly, the best two solutions are always selected, using a simple linear ranking selection. In this method, chromosomes of a population are sorted according to their fitness values [19].

The Fitness function that is proposed has two objectives. The first is to increase the classification performance and the second is to reduce the number of selected features. Even though both parameters are crucial to increase the performance of a machine learning model, classification performance is more important than the number of selected features. Considering the previous, a Fitness function is proposed.

$$Fitness(x, y, z) = y * .97 + (100 - ((\frac{100}{z}) * x)) * .03 \quad (7)$$

Where:

x: Is the selected characteristics

y: Is the classification performance

z: Is the total number of features

The values 0.97 and 0.3 were assigned in order to give a greater weight to higher classification rates and less weight to dimensionality reduction.

The crossover operator is a one-point crossover, the spot to make the crossover is exactly or near in the middle of the chromosome.

The mutation operator, mutates all the population excluding the selected parents, the mutation probability is 0.18 for each allele in the solution.

6 Experimental Results

Along the experimental phase, five datasets taken from the UCI Machine Learning Repository [20] were used. Specifically, the Breast Cancer, Heart disease, Hepatitis diagnosis, Lung Cancer and Arrhythmia data sets were chosen. The characteristics of these datasets are shown in Table 1.

Four algorithms were proposed to make a comparison with the DGA:

Table 2: Mutation rates.

Algorithm	Mutation rate
SGA1	4%
SGA2	26%
SGA3	6%
SGA4	27%

- Simple Genetic Algorithm with Binary Tournament selection method and Singlepoint crossover method called SGA1.
- Simple Genetic Algorithm with Stochastic Remainder Sampling selection method and Singlepoint crossover method called SGA2.
- Simple Genetic Algorithm with Binary Tournament selection method and Doublepoint crossover method called SGA3.
- Simple Genetic Algorithm with Stochastic Remainder Sampling selection method and Doublepoint crossover method called SGA4.

Different mutation rates were used in each algorithm in order to improve performace of each one. Mutation rate of each algorithm is shown in Table 2.

Each algorithm was run 20 times. The search process stopped when the number of evaluated indivuduals got 5% of the solution space in Breast Cancer, Heart and Hepatitis datasets. In Lung Cancer and Arrhythmia datasets the search process stopped when the number of evaluated indivuduals was 100.

Table 3 shows classification performance and dimensionality reduction in each dataset. Near Optimum indicates how far was the solution obtained by genetic algorithms compared to the optimal solution, which is obtained by brute force. After exhaustive exploration of the solution space, the selected subset of features that maximizes classification accuracy is selected. In case that two or more subset of features achieve the same classification accuracy, the subset of features with the smallest number of them is selected.

7 Conclusions

This study presents a genetic algorithm-based approach called DGA. The classical GA operators have been taken and mixed, in a new way to solve a known problem, the feature selection. The results showed that this approach to feature selection obtains an approximately optimal solution in less time than an exhaustive search would need; especially so on data sets where the amount of features starts to prohibit full exploration of the solution space. This suggests that the DGA is a useful tool for feature selection, especially for larger datasets.

The results of the conducted experiments show that the DGA coupled with an associative classifier exhibits a good performance using a reduced solution space.

Table 3: Classification accuracy using 10 fold cross-validation.

		Arrhythmia	Breast Cancer	Heart	Hepatitis	Lung Cancer
DGA	CA	**57.95%**	**96.90%**	**80.15%**	**84.10%**	**81.30%**
	DR	53.53%	41.11%	72.31%	**73.95%**	**71.52%**
	NO		**100%**	**97.39%**	**99.46%**	
SGA1	CA	55.60%	96.45%	73.15%	78.15%	79.05%
	DR	**54.16%**	**46.11%**	**75.38%**	60.53%	64.38%
	NO		85.41%	75.52%	94.38%	
SGA2	CA	51.50%	96.20%	70.40%	68.30%	71.60%
	DR	49.68%	42.78%	51.15%	54.21%	51.07%
	NO		85.41%	66.14%	59.62%	
SGA3	CA	54.45%	96.75%	75.15%	77.25%	78.90%
	DR	52.03%	40%	67.31%	62.89%	59.91%
	NO		85.41%	82.29%	91.71%	
SGA4	CA	51.65%	96.20%	70.90%	68.05%	72.85%
	DR	49.87%	41.11%	53.46%	53.68%	50.80%
	NO		85.41%	69.27%	59.62%	

CA: Classification accuracy, DR: Dimensionality reduction, NO: Near Optimum

The experimental results suggest that it is a feasible approach for datasets consisting of more than a dozen features, obtaining good classification performance values with an important feature amount reduction evaluating a little number of solutions.

It is clear from these results that the DGA is capable of approximating the optimal feature subset within less of 5% in average sized datasets. Regarding to dimensionality reduction, the results show important cut down on the number of features selected. For bigger datasets, a reduction of half or more of the features can be observed.

The key result observed in these experiments is the reduction of the searching space. It can be seen from the result tables that even the most inefficient experiments (the ones on the Breast Cancer dataset) required less evaluations than an exhaustive search to find the optimal solution or one extremely similar to it in performance.

Further work can be done trying to improve the operators used to get an even better solution. Additionally, different genetic operators can be used to find the combination of these which yields better results. Other kinds of evolutionary algorithms should be tested in the future in order to determine if they can produce similar or better results than the DGA; similarly, future research should also focus on coupling the DGA with other classifiers and prove its efficiency.

Also a future research should be done in order to know, the percentage needed to be evaluated in the solution space, to get an almost optimal solution, in big data sets.

References

1. Aldape-Pérez, M., Yánez-Márquez, C., Camacho-Nieto, O., Argüelles-Cruz, A.J.: An associative memory approach to medical decision support systems. Computer Methods and Programs in Biomedicine **106** (May 2012) 287–307
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182
3. Siedlecki, W.W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters **10**(5) (1989) 335–347
4. Kuncheva, L.I.: Fitness functions in editing k-nn reference set by genetic algorithms. Pattern Recognition **30**(6) (1997) 1041–1049
5. Kuncheva, L., Jain, L.C.: Designing classifier fusion systems by genetic algorithms. IEEE Trans. Evolutionary Computation **4**(4) (2000) 327–336
6. Aldape-Pérez, M., Yáñez-Márquez, C., Camacho-Nieto, O., Ferreira-Santiago, Á.: Feature selection using associative memory paradigm and parallel computing. Computación y Sistemas **17**(1) (2013) 41–52
7. Marinakis, Y., Dounias, G., Jantzen, J.: Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification. Computers in Biology and Medicine **39**(1) (2009) 69 – 78
8. Koza, J.R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M.H., Goldberg, D.E., Iba, H., Riolo, R.L.: Genetic programming 1998: Proceedings of the third annual conference. IEEE Trans. Evolutionary Computation **3**(2) (1999) 159–161
9. Nayeem, M.A., Rahman, M.K., Rahman, M.S.: Transit network design by genetic algorithm with elitism. Transportation Research Part C: Emerging Technologies **46** (2014) 30 – 45
10. de Oliveira, R.A., de Medeiros Júnior, M.F., Menezes, R.F.A.: Application of genetic algorithm for optimization on projects of public illumination. Electric Power Systems Research **117** (2014) 84 – 93
11. Uysal, A.K., Gunal, S.: Text classification using genetic algorithm oriented latent semantic features. Expert Systems with Applications **41**(13) (2014) 5938 – 5947
12. Burke, D.S., Jong, K.A.D., Grefenstette, J.J., Ramsey, C.L., Wu, A.S.: Putting more genetics into genetic algorithms. Evolutionary Computation **6**(4) (1998) 387–410
13. Schultz, A.C., Grefenstette, J.J., Jong, K.A.D.: Test and evaluation by genetic algorithms. IEEE Expert **8**(5) (1993) 9–14
14. Jong, K.A.D.: Evolutionary computation - a unified approach. MIT Press (2006)
15. Acevedo-Mosqueda, M.E., Yáñez-Márquez, C., López-Yáñez, I.: Alpha-beta bidirectional associative memories: theory and applications. Neural Processing Letters **26**(1) (2007) 1–40
16. Steinbuch, K.: Adaptive networks using learning matrices. Kybernetik **2**(4) (1964) 148–152

17. Steinbuch, K., Piske, U.A.W.: Learning matrices and their applications. IEEE Trans. Electronic Computers **EC-12**(6) (Dec. 1963) 846–862
18. Kohonen, T.: Correlation matrix memories. IEEE Transactions on Computers **C-21**(4) (1972) 353–359
19. Handels, H., RoSS, T., Kreusch, J., Wolff, H., Pöppl, S.: Feature selection for optimized skin tumor recognition using genetic algorithms. Artificial Intelligence in Medicine **16**(3) (1999) 283 – 297 1999.pdf.
20. Asuncion, A., Newman, D.: UCI Machine Learning Repository (2007)

Reviewing Committee

Magdalena Marciano Melchor
Hind Taud
Itzamá López Yáñez
Miguel G. Villarreal Cervantes
Oscar Camacho Nieto
Edgar Roberto Ramos Silvestre
Hildeberto Jardón Kojakhmetov
William De La Cruz De Los Santos
Jamal Toutouh El Alamin
John Cortes Romero
Alexander Gelbukh
Mateo Valero Cortés
Mónica Isela Acuautla Meneses
Najib Tounsi
Roberto Sepúlveda Lima
Víctor Manuel Ramírez Rivera
Martha Dunia Delgado Dapena
Cornelio Yáñez Márquez
Cuauhtémoc López Martín
Yenny Villuendas Rey
Alfredo Víctor Mantilla Caeiros
Edgar Omar López Caudana
Gerardo Abel Laguna Sánchez
Hiram Calvo Castro
José Fermi Guerrero Castellanos
Julián Patiño Ortiz
Pedro Sánchez Santana
Ollin Peñaloza Mejía
Stephane Couturier
Néstor Velasco Bermeo
Mario Aldape Pérez
Luis Octavio López Leyva
Alejandro Rosete Suárez
Giovanni Guzmán Lugo
Roberto Zagal Flores
David Ortega Pacheco
Gerardo Abel Laguna Sánchez
Elena Acevedo Mosqueda
Rolando Flores Carapia
Miguel Patiño Ortiz
Antonio Hernández Zavala
Benjamín Luna Benoso

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, Ciudad de México.
octubre de 2016
Printing 500 / Edición 500 ejemplares

ISSN: 1870-4069
http://rcs.cic.ipn.mx

RCS
Research in Computing Science