

Aprendizaje máquina y ambientes inteligentes

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

María Fernanda Ríos Zacarias

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 92**, mayo 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el 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, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 92**, May 2015. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Aprendizaje máquina y ambientes inteligentes

Miguel González Mendoza (ed.)



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



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2015

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, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

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

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Editorial

El propósito de este volumen de la revista RCS es reflejar las nuevas direcciones de investigación y aplicaciones de los métodos de la Inteligencia Artificial.

Los artículos de este volumen fueron seleccionados con base en un estricto proceso de revisión efectuada por los miembros del Comité de revisión, tomando en cuenta la originalidad, aportación y calidad técnica de los mismos. Cada artículo fue revisado por lo menos por dos miembros del Comité de revisión del volumen.

Este volumen contiene 13 artículos relacionados con varios aspectos del desarrollo de los métodos de aprendizaje máquina, teniendo como ejemplos de sus aplicaciones en ambientes inteligentes tales como:

- opinión en blogs financieros,
- ambientes inteligentes para el aprendizaje,
- terapias basadas en juegos para adultos mayores,
- monitoreo de emociones para adultos mayores,
- generación de señales en rehabilitación neuromuscular de tobillo,
- sistemas de asistencia alimentaria,
- control de calefacción en invernaderos, entre otros.

Este volumen puede ser interesante para los investigadores y estudiantes de las ciencias de la computación, especialmente en áreas relacionadas con la inteligencia artificial y su aplicación a los diferentes ámbitos de la vida cotidiana; así como, para el público en general interesado en estos fascinantes temas.

En este número especial de la revista RCS, a nombre de la comunidad académica del Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC) y de la SMIA expresamos nuestro agradecimiento al Dr. Sergio Carrera Riva Palacio, Director Ejecutivo, y Dr. Juan Carlos Téllez Mosqueda, Director Adjunto de Innovación y Conocimiento, por apoyar de manera ingente la investigación y el desarrollo de la ciencia y la tecnología, sustentado todo ello en la responsabilidad y el compromiso social.

El proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair, www.EasyChair.org.

Miguel González Mendoza
Mayo 2015

Table of Contents

	Page
A comparison between SARIMA models and Feed Forward Neural Network Ensemble Models for Time Series Data	9
<i>Daniel Alba, Angel Eduardo Muñoz-Zavala</i>	
Árbol de decisión C4.5 basado en entropía minoritaria para clasificación de conjuntos de datos no balanceados	23
<i>Luis Alberto Caballero Cruz, Asdrubal López-Chau, Jorge Bautista López</i>	
Segmentación del padrón de sujetos vulnerables en el Sistema de Asistencia Alimentaria	35
<i>Juan Pablo Granados Garcia</i>	
Control difuso del parámetro β de una heurística constructiva tipo GRASP para el problema de la bisección de vértices de un grafo.....	49
<i>Javier Alberto Rangél González, Jesús David Terán Villanueva, Héctor Joaquín Fraire Huacuja, Juan Javier González Barbosa, Jose Antonio Martínez Flores, Guadalupe Castilla Valdez, Apolinar Ramírez Saldivar</i>	
Generación de señales de electroestimulación percutánea para terapias pasivas en rehabilitación neuromuscular de tobillo empleando lógica difusa	59
<i>Miriam Reyes-Fernandez, Ruben Posada-Gomez, Albino Martinez-Sibaja, Alberto Aguilar-Laserre, Juan Trujillo-Caballero</i>	
Modelado difuso Takagi-Sugeno para sintonizar un controlador de calefacción en invernaderos.....	69
<i>Julio César Ramos Fernández, Marco Antonio Márquez Vera, Luis Fidel Cerecero Natale, Frédéric Lafont, Jean François Balmat</i>	
Monitoreo de emociones aplicadas a terapias basadas en juegos y lógica difusa para adultos mayores.....	81
<i>Monica Janneth Guido Silva, Víctor Manuel Zamudio Rodríguez, Héctor José Puga Soberanes, Juan Martín Carpio Valadez, María Del Rosario Baltazar Flores</i>	
¿La Explicación explica la Abducción? Encuentros y desencuentros entre Inteligencia Artificial y Filosofía de la Ciencia	91
<i>David Gaytán</i>	
Minería de opinión en blogs financieros para la predicción de tendencias en mercados bursátiles	101
<i>Sergio Hernandez, Sabino Miranda-Jimenez, Elio Villaseñor, Eric Téllez, Mario Graff</i>	

Ambiente inteligente de aprendizaje con manejo afectivo para Java.....	111
<i>Ramón Zatarain Cabada, Maria Lucia Barrón Estrada, Francisco Gonzalez Hernandez, Raul Oramas Bustillos</i>	
Ambiente inteligente y afectivo de aprendizaje para compiladores.....	123
<i>María Lucia Barrón Estrada, Ramón Zatarain, Hugo Enrique Gómez Pérez</i>	
An intelligent hospital device for health caring assistance of patients with motor disabilities	133
<i>Eduardo Vazquez-Santacruz, Mariano Gamboa-Zúñiga</i>	
Captura de datos para análisis de la dinámica del tecleo de números para sistema operativo Android	147
<i>Selene Nieto-Ruiz, Yonic Antonio Gomez Sanchez, Asdrubal López-Chau, Carlos A. Rojas</i>	

A Comparison between SARIMA Models and Feed Forward Neural Network Ensemble Models for Time Series Data

Daniel Alba-Cuéllar, Angel Eduardo Muñoz-Zavala

Universidad Autónoma de Aguascalientes,
Aguascalientes, México

albacd@cimat.mx, aemzmx@gmail.com

Abstract. In this paper, we investigate the robustness of Feed Forward Neural Network (FFNN) ensemble models applied to quarterly time series forecasting tasks, by comparing their prediction ability with that of Seasonal Auto-regressive Integrated Moving Average (SARIMA) models. We obtained adequate SARIMA models which required statistical knowledge and considerable effort. On the other hand, FFNN ensemble models were readily constructed from a single FFNN template, and they produced competitive forecasts, at the level of well-constructed SARIMA models. The single template approach for adapting FFNN ensembles to multiple time series datasets can be an economic and sensible alternative if fitting individual models for each time series turns out to be very time consuming. Additionally, FFNN ensembles were able to produce accurate interval estimations, in addition to good point forecasts.

Keywords: time series forecasts, SARIMA models, artificial neural networks, model ensemble, particle swarm optimization, ensemble channel.

1 Introduction

A central objective pursued by people working in diverse scientific and engineering fields is to predict, as accurately as possible, short-term future behavior from relatively well-understood, but out of control time-evolving variables, like birth rates in a municipality, employment rates in a national economy, enrollments at a certain school, influenza cases detected in a community, blood pressure measured from an individual, rainfall at a specific geographic region, etc. Reasonable short-term predictions for variables such as these enable people to make informed decisions; for instance, good 24-hour-ahead predictions for electrical energy consumption are of fundamental importance when deciding about the actions to take in order to guarantee the optimal management and operation of an energy grid supplying electricity to clients.

A time series is a sequence of chronologically ordered values, sampled from a time-evolving variable; this conceptual tool plays a central role in the scientific

search for good short-term predictions (also called forecasts). Typically, time series values are measured from a real-world process, or variable, at equally-spaced points in time. In order to predict future time series values, mathematical models are typically employed; these models extract information about the characteristic behavior of consecutive past time series values (historical data), and then use this extracted information to project (extrapolate) time series behavior into the near future.

We can identify two broad classes of mathematical methodologies for building time series models: traditional methods based on parametric statistical modeling, and newer techniques based on nonparametric modeling. One potential limitation of parametric models is that they make strong assumptions about the true nature of the time series generating mechanism (e.g., temporal data are generated by a linear, time-invariant process). This limitation, coupled with the advent of powerful and cheap computing devices, motivated researchers to consider more flexible modeling strategies: enter the nonparametric approach, which includes support vector regression models [5]; artificial neural network (ANN) models [10], [11]; and wavelet methods [6].

Although ANNs are often employed for supervised classification and pattern recognition tasks, it was soon realized that ANNs are also a good alternative to regression problems, and more specifically, to time series forecasting tasks. Lapedes reports the first attempt to model nonlinear time series with artificial neural networks [9]. ANNs achieve universal functional approximation; it has been shown that an artificial neural network can approximate any continuous function (linear or nonlinear) to any desired accuracy [4]. Time series modeling, in particular, is a function approximation problem, so ANNs seem to be a natural alternative to this kind of problem. Feed Forward Neural Network (FFNN) models belong to a particular class of ANNs; they combine sigmodal activation functions in order to achieve nonlinear mappings.

In this paper, we'll investigate the robustness of FFNN ensembles applied to quarterly time series forecasting tasks, by comparing their prediction ability with that of SARIMA models (SARIMA is a class of ARIMA model which considers seasonal fluctuations present in temporal data). The rest of this document is organized as follows: Sect. 2 describes briefly the main theoretical concepts used in this experiment to construct our models: Time Series modeling basics, ARIMA models, FFNNs, and the basic PSO algorithm; Sect. 3 describes the experimental methodology; Sect. 4 describes the results of our experiment, and Sect. 5 summarizes our findings and conclusions.

2 Conceptual Tools

2.1 Time Series Modeling: the Problem of Predicting the Future

Extending backwards from time t , we have a time series $\{y_t, y_{t-1}, y_{t-2}, \dots\}$. Employing this information, we now want to estimate y at some future time $t + s$ (s is called the prediction horizon; typically, $s = 1$). We can accomplish

this task if we assume a model in which y_{t+s} is generated from a functional relationship f involving past y values $y_t, y_{t-1}, y_{t-2}, \dots, y_{t-d+1}$; i.e.,

$$y_{t+s} = f(y_t, y_{t-1}, y_{t-2}, \dots, y_{t-d+1}). \quad (1)$$

This is a function approximation problem. To solve it, we'll typically go through the following steps: 1. Assume a generative model f ; 2. For every time point t_p in the past, train f using $y_{t_p}, y_{t_p-1}, y_{t_p-2}, \dots, y_{t_p-d+1}$ as **inputs** and y_{t_p+s} as the **target** (this step is called *training phase*); 3. Now run the trained generative model f to predict y_{t+s} from $y_t, y_{t-1}, y_{t-2}, \dots, y_{t-d+1}$. This procedure, characterized by functional relationship (1), is called *auto-regressive approach to time series modeling*. The objective of the training phase is to adjust the behavior (parameters) of f until all of its predictions \hat{y}_{t_p+s} get sufficiently close to corresponding target values y_{t_p+s} . A properly trained generative model f will produce forecasts \hat{y}_{t+s} reasonably close to future time series values y_{t+s} ; we say in this case that the model f generalizes well to unknown future time series values.

2.2 ARIMA Time Series Modeling

Perhaps the most popular and traditional of all statistical methods for time series forecasting is *auto-regressive integrated moving average* (ARIMA) modeling. The general ARMA model (without the I) was described in [19], and it was popularized in the 1970 book by [3]; it has the form

$$y_t = \sum_{i=1}^p \phi_i y_{t-i} + \sum_{j=1}^q \theta_j w_{t-j} + w_t. \quad (2)$$

Auto-Regressive (AR) parameters ϕ_i and Moving Average (MA) parameters θ_j in (2) are estimated from the time series historical part (training data), usually via maximum likelihood; in this way, we obtain maximum likelihood estimates $\hat{\phi}_i$ and $\hat{\theta}_j$. White noise terms w_t are independent, identically distributed random variates which come from a normal distribution $N(0, \sigma_w^2)$. To estimate future values of y_t , we generally assume that w_t at time t is small relative to y_t . We can obtain estimates of past values of w_t at time $t-i$ from past true values of y_t and past values of \hat{y}_t : $\hat{w}_{t-i} = y_{t-i} - \hat{y}_{t-i}$; the estimate for y_t is then

$$\hat{y}_t = \sum_{i=1}^p \hat{\phi}_i y_{t-i} + \sum_{j=1}^q \hat{\theta}_j \hat{w}_{t-j}. \quad (3)$$

Extensions to the basic ARMA model include ARIMA models for dealing with non-stationary time series with trend, and SARIMA models, which are useful to identify seasonal patterns in temporal data; such seasonal patterns are characterized in a SARIMA model by Seasonal Auto-Regressive (SAR) and Seasonal Moving Average (SMA) parameters. For more information, see [13].

2.3 Feed Forward Neural Networks

An artificial neural network can be graphically represented as a group of nodes interconnected by arrows; such representation helps us visualize how information is processed inside a system of artificial neurons. We now describe the structure of a feed forward neural network (FFNN) adapted to our experiment. Figure 1 depicts a FFNN with inputs x_1, x_2, \dots, x_d and output \hat{y} .

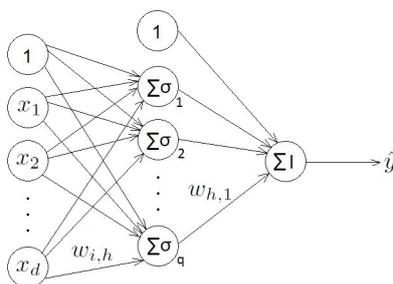


Fig. 1. Feed Forward Neural Network (FFNN)

In the structure from Fig. 1, information flows from left to right. Each arrow inside of the neural network represents a weight w ; a signal with value s at the left of arrow w , flows through it, and exits at the right side with value $w \cdot s$. A feed forward neural network (also called a multilayer perceptron) typically contains three layers: input layer, hidden layer, and output layer. The input layer consists simply of the network's inputs; i.e., x_1, x_2, \dots, x_d (predictor values), plus a constant value of one called intercept. In Fig. 1, our input layer consists of the nodes (circles) aligned vertically on the left side. The column of nodes at the immediate right of the input layer is the hidden layer, which consists of q neurons, or hidden units, placed at the nodes, plus an intercept. Each hidden unit sums all of its inputs, and then transforms this sum via its activation function $\sigma(z) = \frac{1}{1+e^{-z}}$ (sigmoidal logistic function). Mathematically, the functionality of hidden neuron h ($h = 1, 2, \dots, q$) is described by $\sigma\left(\sum_{i=0}^d w_{i,h}x_i\right)$, where $x_0 = 1$ is the intercept coming from the input layer; $w_{i,h}$ is the weight that corresponds to the arrow connecting input node x_i to hidden unit h . The output layer in Fig. 1 consists of one neuron only (since we are interested in forecasting a single time series value), with similar functionality to that of hidden neurons; the difference is that now the activation function has the form $I(z) = z$ (identity function). The functionality of the output layer (and of the whole feed forward neural network) is described by

$$\hat{y} = w_{0,1} + \sum_{h=1}^q w_{h,1}\sigma\left(\sum_{i=0}^d w_{i,h}x_i\right), \quad (4)$$

where $w_{0,1}$ is the weight that corresponds to the arrow connecting the hidden layer intercept to the output unit, and $w_{h,1}$ is the weight that corresponds to the arrow connecting hidden neuron h to output unit.

FFNNs are trained by using supervised machine learning algorithms, Back-Propagation (BP) being the most representative of such algorithms for FFNN training purposes. BP attempts to minimize a loss function which involves all weights in the FFNN; a typical loss function employed is the mean squared error loss function

$$E(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^n \left[y^{(i)} - \hat{y}^{(i)}(\mathbf{W}) \right]^2, \quad (5)$$

where $y^{(i)}$ is the target (observed value) and $\hat{y}^{(i)}(\mathbf{W})$ is the FFNN output for the i -th example input pattern in the training data ($i = 1, 2, \dots, n$); \mathbf{W} is a vector whose components are all the weights in the FFNN. The objective of BP is to adjust the weights in \mathbf{W} iteratively, using a stochastic gradient descent technique, until the FFNN output gets sufficiently close to all target values in the training data set. For more details, see [2].

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [8] is an optimization algorithm inspired by the motion of a bird flock; any member of the flock is called a “particle”. All particles move across a real valued D -dimensional search space, so each particle has three attributes: position \mathbf{x} , velocity \mathbf{v} , and best position visited after the first iteration \mathbf{p} , according to a cost function E . The best of all \mathbf{p} values is called global best \mathbf{g} ; this global best position is communicated to all particles such that, at the time of the next algorithm iteration, all the particles are aware of the best position visited. PSO stated goal is to minimize function $E : \mathbb{R}^D \rightarrow \mathbb{R}$; i.e., find $\mathbf{a} \in \mathbb{R}^D$ such that $E(\mathbf{a}) \leq E(\mathbf{x})$ for all \mathbf{x} in the search space. Figure 2 lists a pseudo-code for the basic PSO algorithm; here, ω is the inertia weight, and φ_p, φ_g are called acceleration coefficients; these meta-parameters must be adjusted by the practitioner.

Recently, PSO has been employed as a FFNN training algorithm; it is easy to adapt the basic PSO algorithm in order to minimize mean squared error loss function (5); simply identify vector \mathbf{W} , whose components are all the weights in a FFNN, with basic PSO’s position vector \mathbf{x} . Numerous particular applications involving some sort of FFNN-PSO hybrid model for time series prediction can be found in the literature; see for example [1], [12] and [20]. In this paper, we adapt FFNN ensemble models to quarterly time series data, training individual FFNN elements with either BP or basic PSO; individual predictions from an ensemble are averaged out, producing a final prediction for the whole ensemble. Individual predictions taken together, but not combined, form an ensemble channel. See Sect. 3 for more information.

Algorithm 1: basic PSO algorithm

```

for each particle  $i = 1, 2, \dots, N$  in the swarm do
  initialize position:  $\mathbf{x}_i \leftarrow$  uniform random vector in  $\mathbb{R}^D$ 
  initialize best known position:  $\mathbf{p}_i \leftarrow \mathbf{x}_i$ 
  if  $E(\mathbf{p}_i) < E(\mathbf{g})$  update swarm's best known position:  $\mathbf{g} \leftarrow \mathbf{p}_i$ 
  initialize velocity:  $\mathbf{v}_i \leftarrow$  uniform random vector in  $\mathbb{R}^D$ 
until a termination criterion is met, repeat:
  for each particle  $i = 1, 2, \dots, N$  in the swarm do
    for each dimension  $d = 1, 2, \dots, D$  do
      pick random numbers  $r_p, r_g \sim U(0, 1)$ 
      update velocity:  $v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p r_p (p_{i,d} - x_{i,d}) + \varphi_g r_g (g_d - x_{i,d})$ 
      update position :  $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$ 
    if  $E(\mathbf{x}_i) < E(\mathbf{p}_i)$  do
      update best known position:  $\mathbf{p}_i \leftarrow \mathbf{x}_i$ 
      if  $E(\mathbf{p}_i) < E(\mathbf{g})$  update swarm's best known position:  $\mathbf{g} \leftarrow \mathbf{p}_i$ 
Now  $\mathbf{g}$  holds the best found solution

```

Fig. 2. Basic Particle Swarm Optimization (PSO) algorithm.

3 Experimental Methodology

The objective of this experiment is to investigate the robustness of FFNN ensembles applied to quarterly time series forecasting tasks, by comparing their prediction ability with that of SARIMA models. For constructing our models, we employed eleven time series datasets taken from the 2010 Neural Network Grand Competition (NNGC) [16]. The time series datasets employed in this experiment are listed in Table 1. Next we'll describe the conditions under which the experiment was conducted.

Table 1. Time series datasets employed in this experiment. Series with a mark * are seasonally adjusted.

Time series name	from	to	Quarterly values
1. Total operating expenses for US domestic airlines	1993-I	2002-IV	40
2. Personal Expenditures: transportation	1990-I	1997-III	31
3. Motor vehicle output (Autos)	1967-I	2003-IV	148
4. Motor vehicle output (Trucks)	1967-I	2003-IV	148
5. New motor vehicles (Trucks)	1967-I	2003-IV	148
6. Net exports (Autos)	1977-I	2003-IV	108
7. Net exports (Trucks)	1977-I	2003-IV	108
8. Final sales of vehicles to domestic purchasers	1967-I	2003-IV	148
9. Domestic output of new autos	1967-I	2003-IV	148
10. Real Vehicle Output-Private fixed investment *	1967-I	2003-IV	148
11. Real Vehicle Output-Imports *	1967-I	2003-IV	148

For the SARIMA models we employed the basic Box-Jenkins methodology, fitting models of the form $ARIMA(p, d, q) \times (P, D, Q)_4$ to each time series dataset. On the other hand, for FFNN ensembles, we fit a single fixed FFNN structure to each FFNN element in any given ensemble, across all time series datasets.

Prior to commencing model construction, we standardized each time series dataset (i.e., we subtracted dataset mean from each observation and divided resulting difference by dataset standard error); we did this in an initial attempt to minimize the occurrence of convergence problems inside model training procedures. During the construction of SARIMA models, we realized that series 4, 5, 7 and 8 show growing variability (heteroscedasticity), so first we applied a logarithmic transformation to the original series in those cases, and then we standardized the log transformed series. The forecasts produced by all constructed models were compared against transformed target values. In this experiment we considered two types of ensembles: FFNN-BP ensembles, and FFNN-PSO ensembles; all individual models in a given ensemble were trained using just one of the two training algorithms considered in this experiment: Back-Propagation, and PSO.

Model implementation. All models in this experiment were implemented by using the R programming language [17]. SARIMA models were implemented with the help of the `astsa` package [14]; individual FFNN-BP models in an ensemble were implemented via the `nnet` package [15], and FFNN-PSO models were programmed from scratch, using a combination of R and Java code. For FFNN-PSO ensembles, the programming of the FFNN part, was done in R, while the PSO part was done in Java, in order to improve computing speed. R and Java code communicate inside the R environment via the `rJava` package [18]. The basic PSO algorithm, according to Kennedy and Eberhart [8] was the algorithm of choice in our experiment. For FFNN models, we chose a 4-2-1 FFNN architecture (i.e., 4 input variables, 2 hidden sigmoid units and 1 output linear unit). This choice of FFNN size was made taking into account the parsimony principle for model construction: we chose the minimal number of input nodes capable of capturing annual seasonal patterns for quarterly time series, and a small number of hidden units so as to allow for a moderate amount of nonlinearity. One output unit is all we need, because we are producing 1-step ahead forecasts for univariate time series.

Notational conventions for model building. Suppose a given time series dataset contains N (normalized) values x_1, x_2, \dots, x_N equally spaced in time. For all FFNN models considered in this experiment, inputs are time series values $x_t, x_{t-1}, x_{t-2}, x_{t-3}$, while for all models (including SARIMA models), the corresponding output \hat{x}_{t+1} is considered to be an estimate of value x_{t+1} . Now we'll describe how we split the time series datasets into training and test sets.

Training set. It consists of all values in a given time series dataset, except for the 4 most recent, which are reserved for testing purposes. For FFNN models, it is necessary to rearrange training set values into a rectangular array like the one shown in Table 2. Columns $x_{t-3}, x_{t-2}, x_{t-1}, x_t$ represent predictor variables, while column x_{t+1} contains target values.

Test set. It consists of the 4 most recent observations $x_{N-3}, x_{N-2}, x_{N-1}, x_N$ in a given time series dataset. These values are to be compared against model predictions $\hat{x}_{N-3}, \hat{x}_{N-2}, \hat{x}_{N-1}, \hat{x}_N$, so we can measure forecast ability for any model; see Equation (6).

Table 2. Training data for FFNN models.

Example	x_{t-3}	x_{t-2}	x_{t-1}	x_t	x_{t+1}
1	x_1	x_2	x_3	x_4	x_5
2	x_2	x_3	x_4	x_5	x_6
3	x_3	x_4	x_5	x_6	x_7
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$N - 8$	x_{N-8}	x_{N-7}	x_{N-6}	x_{N-5}	x_{N-4}

In order to generate forecast values for any trained model, we proceed as follows: First, we compute \hat{x}_{N-3} by using values $x_{N-7}, x_{N-6}, x_{N-5}, x_{N-4}$ as model inputs; then, we generate test forecast value \hat{x}_{N-2} by using values $x_{N-6}, x_{N-5}, x_{N-4}, \hat{x}_{N-3}$ as model inputs; \hat{x}_{N-1} is generated from inputs $x_{N-5}, x_{N-4}, \hat{x}_{N-3}, \hat{x}_{N-2}$, and finally \hat{x}_N is generated from inputs $x_{N-4}, \hat{x}_{N-3}, \hat{x}_{N-2}, \hat{x}_{N-1}$; notice that each succeeding forecast employs more and more predicted values as inputs, so forecast uncertainty grows as we advance into the distant future.

Test error. To measure forecast ability for any model considered in this experiment, we employed a Mean Squared Error (MSE) measure, computed according to Equation (6).

$$MSE = \frac{\sum_{t=N-3}^N (x_t - \hat{x}_t)^2}{4}. \quad (6)$$

FFNN model training. As previously mentioned, a fixed FFNN architecture 4-2-1 with a linear output unit and sigmoidal hidden units was chosen. FFNN-BP models were implemented and trained adjusting weight decay to 0.01, and maximum number of iterations to 1000. These settings were applied to each individual FFNN-BP element in an ensemble, across the 11 time series datasets.

FFNN-PSO models in this experiment employed 100 particles (each particle conceptually being a FFNN), evolving through 200 iterations; particle position components (which happen to be FFNN weights) were randomly initialized using a uniform distribution $U(-2, 2)$; likewise, particle velocity components were randomly initialized, but using a uniform distribution $U(-0.7, 0.7)$. If any particle velocity component v exceeds 1.4 in absolute value at any iteration of the PSO process, it is set back to 0.7, multiplied by the sign of v ; this is done to prevent particles from moving too fast in the search space, thus favoring a stable evolution of the PSO process. As for the rest of PSO meta-parameters, the inertia weight ω was set equal to 1.5, while acceleration coefficients φ_p and φ_g were both set equal to 2.0 (see Fig. 2). Why these choices for our FFNN-PSO models? Position and velocity components were randomly initialized with rather small values, as this provides better convergence behavior. In previous experiments, we observed that 10 or even 50 particles were not sufficient for proper convergence; on the other hand, more than 100 particles take too much computation time and produce no better results than 100 particles. Previous experimentation also

shows that 200 iterations are enough for our PSO algorithm to converge; in this experiment, FFNN-PSO training stops exactly after 200 iterations. Acceleration coefficients were both set to 2.0 and inertia weight was set to 1.5 because we observed that the recommended values of 0.72 for inertia weight and 1.49 for both acceleration coefficients, suggested in the literature (see, for example, [7]), were rather slow for proper convergence.

4 Experimental Results

MSE measures for SARIMA models. Table 3 shows MSE measures, computed as indicated by Equation (6), for constructed SARIMA models, along with their final selected orders $(p, d, q) \times (P, D, Q)_4$; these orders were selected carefully: sample ACF and PACF plots were inspected thoroughly in order to determine if regular and/or seasonal differentiation was required; tentative AR, SAR, MA and SMA orders were tested by fitting several SARIMA models; diagnostics were inspected to make sure residuals from a given model were as uncorrelated and normal as possible; finally, to select a winner from a list of competing SARIMA models adequately fitted to a time series dataset, Akaike and Bayesian Information Criteria were used. Recall from Table 1 that series 10 and 11 are seasonally adjusted, which means that all seasonal influences from these series were already removed; this is why P, Q and D orders for SAR, SMA components and Seasonal differencing all equal zero in these series.

Table 3. MSE measures for constructed SARIMA models. $\text{std}(x)$ means original series was standardized; $\text{std}(\log(x))$ means log of original series was standardized.

Series	SARIMA order	Transformation	MSE
1	$(0, 1, 0) \times (0, 0, 0)_4$	$\text{std}(x)$	0.0239
2	$(0, 1, 0) \times (0, 0, 0)_4$	$\text{std}(x)$	0.0435
3	$(1, 1, 1) \times (0, 0, 0)_4$	$\text{std}(x)$	0.1578
4	$(0, 1, 0) \times (0, 1, 5)_4$	$\text{std}(\log(x))$	0.0006
5	$(0, 1, 3) \times (2, 0, 0)_4$	$\text{std}(\log(x))$	0.0033
6	$(1, 1, 1) \times (2, 0, 2)_4$	$\text{std}(x)$	0.0467
7	$(1, 1, 0) \times (0, 0, 0)_4$	$\text{std}(\log(x))$	0.0030
8	$(1, 1, 0) \times (0, 0, 0)_4$	$\text{std}(\log(x))$	0.0025
9	$(0, 1, 1) \times (0, 0, 3)_4$	$\text{std}(x)$	0.1701
10	$(0, 1, 2) \times (0, 0, 0)_4$	$\text{std}(x)$	0.0403
11	$(0, 1, 0) \times (0, 0, 0)_4$	$\text{std}(x)$	0.0317

SARIMA vs. individual FFNN-PSO models (MSE for test data).

Table 4 shows the distribution of prediction errors associated to the elements in each ensemble of FFNN-PSO models (we constructed one ensemble of 100 elements per time series dataset), along with the prediction error associated to the corresponding SARIMA model. From Table 4, we see that for each one of the 11 time series datasets, there is at least one individual FFNN-PSO model which produces a prediction error smaller than the prediction error associated to the corresponding SARIMA model; this tells us that if we build an ensemble

Table 4. SARIMA vs. individual FFNN-PSO (MSE for test data).

series	MSE SARIMA	min	1st q	median	mean	3rd q	max	se
1	0.0239	0.0169	0.3649	0.4501	0.4455	0.5468	0.7428	0.1552
2	0.0435	0.0371	0.0682	0.1014	0.1441	0.1601	0.7871	0.1376
3	0.1578	0.0042	0.0405	0.0754	0.0821	0.1093	0.2102	0.0501
4	0.0006	0.0001	0.0085	0.0298	0.0812	0.0600	2.1110	0.2321
5	0.0033	0.0012	0.0160	0.0556	0.1021	0.1428	0.5756	0.1179
6	0.0467	0.0406	0.1906	0.3242	0.3690	0.5258	0.9925	0.2287
7	0.0030	0.0028	0.0569	0.0872	0.0885	0.1145	0.2231	0.0456
8	0.0025	0.0005	0.0052	0.0184	0.0419	0.0504	0.3193	0.0612
9	0.1701	0.0206	0.0682	0.1146	0.1155	0.1510	0.2696	0.0573
10	0.0403	0.0184	0.0762	0.1655	0.2164	0.3051	0.7544	0.1742
11	0.0317	0.0096	0.0221	0.0346	0.0773	0.0720	0.7600	0.1224

of FFNN-PSO models with at least 100 elements, some of those elements will produce better forecasts than an adequately-built SARIMA model; additionally, all forecasts from the FFNN-PSO ensemble will form an “ensemble channel” which hopefully will contain all time series test values without being too wide (see Fig. 6). Thus, with a single fixed template employed to construct all FFNN-PSO models in this experiment, we were able to produce competitive FFNN-PSO ensembles, at the level of well-constructed SARIMA models; the single template approach could be an economic alternative if fitting individual models for each time series dataset turns out to be a very time consuming task; of course, better results would be achieved by adapting a FFNN-PSO model template to each time series dataset.

SARIMA vs. individual FFNN-BP models (MSE for test data). We also performed an analogous comparison between prediction errors associated to individual FFNN-BP elements in an ensemble, and to SARIMA models; the results are shown in Table 5. We see that, for time series 1, 2, 4, 5, 6, 7 and 10, individual FFNN-BP models never produce better time series forecasts; by contrast, for series 3, 8, 9 and 11, nearly all individual FFNN-BP models produce better time series forecasts with respect to their SARIMA counterparts. Note also from Table 4 and Table 5, that prediction errors for FFNN-PSO ensembles contain more variability than their FFNN-BP counterparts; the resulting variability induced by the PSO process will allow us to estimate prediction limits (see Fig. 3). Unfortunately, small variability of prediction errors from FFNN-BP ensembles does not help in the construction of ensemble channels (see Fig. 4).

FFNN ensemble channels and SARIMA confidence intervals. Figure 3 shows the averages of individual estimations produced by all elements in the 11 FFNN-PSO ensembles built in this experiment (one average per ensemble), along with the corresponding original time series, and the limits for the ensemble channels; the thick vertical lines shown here separate training data from test data. Figure 4 is the FFNN-BP counterpart of Fig. 3, and Fig. 5 is similar to Fig. 3, but shows instead the SARIMA forecasts along with their corresponding prediction limits, built upon the assumption of residual normality. From Fig. 3 and Fig. 5, we see that most test data for the 11 time series datasets are contained

Table 5. SARIMA vs. individual FFNN-BP (MSE for test data).

series	MSE SARIMA	min	1st q	median	mean	3rd q	max	se
1	0.0239	0.3192	0.5262	0.5306	0.5202	0.5319	0.5956	0.0370
2	0.0435	0.0495	0.0495	0.0521	0.0526	0.0522	0.0859	0.0050
3	0.1578	0.0884	0.1012	0.1012	0.0991	0.1022	0.1034	0.0051
4	0.0006	0.0107	0.0109	0.0110	0.0132	0.0139	0.0203	0.0035
5	0.0033	0.0085	0.0085	0.0092	0.0140	0.0126	0.0347	0.0089
6	0.0467	0.0609	0.0647	0.0647	0.0688	0.0655	0.2049	0.0200
7	0.0030	0.0547	0.0599	0.0608	0.0613	0.0643	0.0644	0.0026
8	0.0025	0.0013	0.0013	0.0014	0.0025	0.0019	0.0070	0.0020
9	0.1701	0.0776	0.0787	0.0798	0.0835	0.0838	0.0981	0.0067
10	0.0403	0.0873	0.0938	0.0970	0.0971	0.0978	0.1176	0.0050
11	0.0317	0.0110	0.0112	0.0112	0.0113	0.0113	0.0117	0.0002

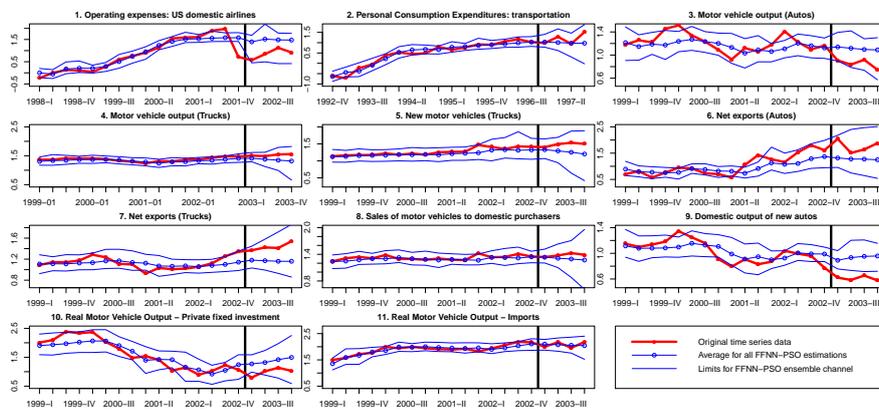


Fig. 3. FFNN-PSO ensemble predictions for all time series datasets.

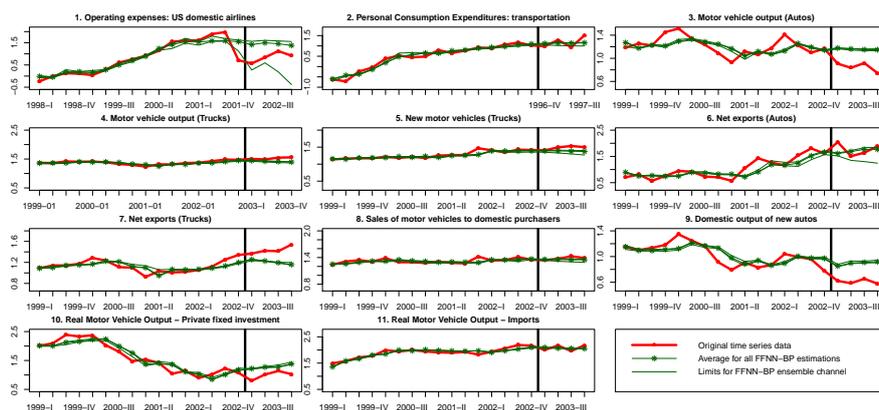


Fig. 4. FFNN-BP ensemble predictions for all time series datasets.

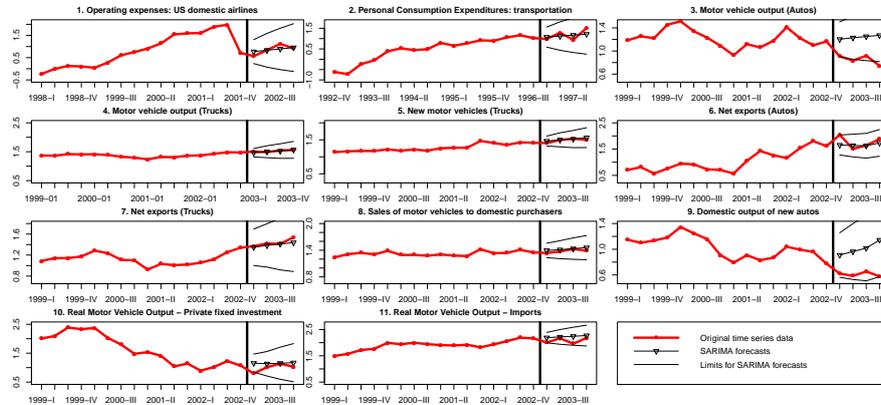


Fig. 5. SARIMA predictions for all time series datasets.

inside the prediction limits defined by the models. To be more precise, SARIMA models enclose 91% of all test data within their prediction limits (indeed getting close to the theoretical expected confidence value of 95%), while FFNN-PSO ensembles enclose 89% of all test data within their ensemble channels; by contrast, FFNN-BP ensembles enclose only 16% of all test data within their prediction limits. Thus, SARIMA models and FFNN-PSO ensembles are able to produce accurate interval estimations, in addition to point forecasts; FFNN-BP ensembles practically give us point forecasts only (except for time series 1 and 6; see Fig. 4). From Fig. 3, we see that FFNN-PSO forecast limits for time series 4, 5 and 8 are too wide; note also that in these cases, ensemble channels enclose training data comfortably, even with a spare margin (by the way, in Fig. 5 we see that SARIMA forecast limits for time series 1, 3, 7 and 9 are also too wide).

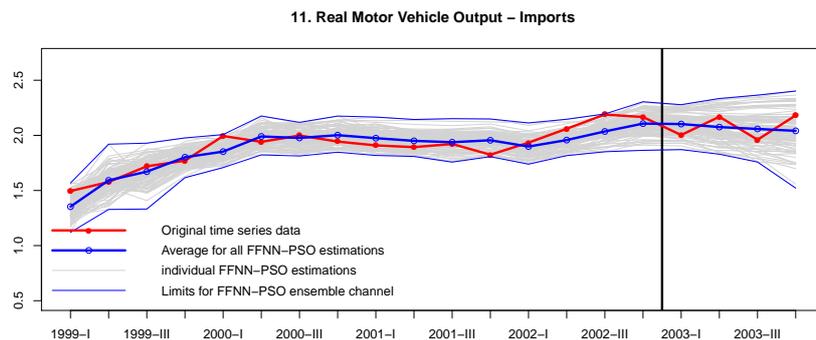


Fig. 6. FFNN-PSO ensemble channel for time series 11.

5 Conclusions

In this experiment, we obtained adequate SARIMA models; their construction required statistical knowledge and good judgment in order to select the best model template for each time series; forecasts produced by these SARIMA models served as a baseline for our model comparison experiment. On the other hand, FFNN ensemble models for all quarterly time series datasets were readily constructed from a single FFNN template, without making any previous assumptions about the data generating mechanism; the point in comparing FFNN ensemble models built from a single template against individual SARIMA models is to test robustness of FFNN ensembles applied to quarterly time series prediction tasks. All individual FFNN elements in an ensemble model were trained using just one optimization algorithm (Back-Propagation or Particle Swarm Optimization) with pre-established meta-parameters, and a fixed training set. Estimations from individual FFNN elements in an ensemble were averaged to form a final ensemble prediction; all individual predictions from the ensemble elements, taken together, but not combined, form an ensemble channel, which suggests multiple possibilities for future time series movements. All FFNN-PSO ensembles generated competitive forecasts, at the level of well-constructed SARIMA models; the single template approach for adapting FFNN-PSO ensembles to multiple time series datasets of quarterly frequency can be an economic and sensible alternative if fitting individual models for each time series dataset turns out to be a very time consuming task. FFNN-PSO ensembles were able to produce accurate interval estimations, in addition to good point forecasts; FFNN-BP ensembles generally produced point forecasts only. In some cases, individual FFNN-BP models produced forecasts more accurate than those produced by SARIMA models or by FFNN-PSO ensembles. Prediction intervals from SARIMA and FFNN-PSO ensemble models covered most test data; in a few cases, prediction intervals for both models were too wide; thus, PSO meta-parameter fine-tuning is necessary in a case-by-case basis. Future work should also include testing FFNNs of varying complexity, to see if forecasting accuracy can be further improved.

Acknowledgments. The first author gratefully acknowledges the financial support from CONACyT to pursue graduate studies at Universidad Autónoma de Aguascalientes. The authors would also like to thank all people involved in the reviewing process for their comments and suggestions which contributed to the improvement of this article.

References

1. Adhikari R., Agrawal R., Kant L.: PSO based Neural Networks vs. traditional statistical models for seasonal time series forecasting. 3rd International Advance Computing Conference (IACC), pp. 719–725. IEEE (2013)
2. Bishop C.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)

3. Box G., Jenkins G.: Time series analysis: forecasting and control. New York, NY: Holden Day (1970)
4. Cybenko G.: Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2(4):303–314 (1989)
5. Drucker H., Burges C., Kaufman L., Smola A., Vapnik V.: Support vector regression machines. *Advances in neural information processing systems*, 9, 155–161 (1997)
6. Härdle W.: *Nonparametric and semiparametric models*. Springer Science & Business Media (2004)
7. Hsieh S., Sun T., Liu C., Tsai S.: Efficient Population Utilization Strategy for Particle Swarm Optimizer. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol 39, Issue 2, IEEE (2009)
8. Kennedy J., Eberhart R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference On Neural Networks*, IEEE, vol. 4, pp 1942–1948 (1995)
9. Lapedes A., Farber R.: Nonlinear signal processing using neural networks: Prediction and system modeling. Tech. Rep. LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM (1987)
10. McCulloch W., Pitts W.: A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133 (1943)
11. Rosenblatt F.: The perceptron - a perceiving and recognizing automaton. Tech. Rep. 85-460-1, Cornell Aeronautical Laboratory (1957)
12. Sheikhan M., Mohammadi N.: Time series prediction using PSO-optimized neural network and hybrid feature selection algorithm for IEEE load data. *Neural Computing and Applications*, Volume 23, Issue 3-4, pp 1185–1194, Springer-Verlag London Limited (2013)
13. Shumway R., Stoffer D.: *Time series analysis and its applications (with R examples)*. 3rd edn, Springer Science+Business Media, LLC (2011)
14. Package “astsa”, <http://www.stat.pitt.edu/stoffer/tsa3/>
15. Package “nnet”, <http://www.stats.ox.ac.uk/pub/MASS4/>
16. Artificial Neural Network & Computational Intelligence Forecasting Competition, <http://www.neural-forecasting-competition.com/instructions.htm>
17. R Core Team: *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org/> (2014)
18. Package “rJava”, <http://www.rforge.net/rJava/>
19. Whittle P.: *Hypothesis Testing in Time Series Analysis*. Hafner Publishing Company (1951)
20. Yeh W.: New Parameter-Free Simplified Swarm Optimization for Artificial Neural Network Training and Its Application in the Prediction of Time Series. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24, Issue 4, pp. 661–665, IEEE (2013)

Árbol de decisión C4.5 basado en entropía minoritaria para clasificación de conjuntos de datos no balanceados

Luis A. Caballero-Cruz, Asdrúbal López-Chau, Jorge Bautista-López

Ingeniería en Computación, Centro Universitario UAEM Zumpango, Zumpango,
Estado de México, México

Resumen. En el área de aprendizaje automático, el problema de desbalance en los datos es uno de los más desafiantes. Desde hace más de una década, se han desarrollado nuevos métodos para mejorar el desempeño de los métodos de clasificación para este tipo de problema. En este artículo se presenta una modificación al algoritmo C4.5 usando el concepto de entropía minoritaria. La propuesta está basada en la corrección de un error que observamos en una publicación previa. La implementación del nuevo método presentado es probada con conjuntos públicamente disponibles. Los resultados obtenidos muestran la utilidad del método desarrollado.

Palabras clave: desbalance, clasificación, entropía minoritaria, árbol de decisión.

1. Introducción

En minería de datos, la tarea de clasificación se refiere al proceso de utilizar un modelo —llamado clasificador— que permita realizar predicciones sobre el tipo, clase o categoría al que pertenecen los objetos que se le presentan a dicho modelo. Las posibles clases de objetos son conocidas de antemano, y el modelo es construido utilizando un conjunto de datos de entrenamiento, en el que los objetos o instancias están previamente etiquetados.

Un problema que se presenta con los métodos de clasificación tradicionales, sucede cuando los datos usados para construir el clasificador se encuentran no balanceados, es decir; cuando las clases o categorías del conjunto de datos de entrenamiento no se encuentran en cantidades aproximadamente proporcionales [8]. Un ejemplo típico ocurre en el análisis financiero, ya que en los datos de las personas que poseen créditos bancarios, la cantidad de personas que presentan adeudo de cantidades muy grandes, es notablemente menor a la cantidad de personas con adeudo de cantidades pequeñas. Otro ejemplo, en un ámbito muy diferente, se puede encontrar con datos de alumnos de nivel escolar primaria en México. La experiencia indica que son más los alumnos que aprueban el ciclo escolar que aquellos que no aprueban. En la literatura, se han detectado muchos otros ejemplos del mundo real donde sucede este tipo de desbalance en los datos.

En este tipo de aplicaciones, es de suma importancia la correcta predicción de instancias que pertenecen a la clase con la menor cantidad de elementos en el conjunto de datos de entrenamiento [6], llamada clase minoritaria.

En este artículo, se propone una modificación al método de clasificación C4.5, que es un árbol de decisión. Esta modificación permite seleccionar, de forma diferente a la tradicional, los puntos de separación de datos que realiza el clasificador de manera recursiva. De esta manera, la mayor parte de los objetos de clase minoritaria son concentrados al centro o en los extremos de las tres particiones que se realiza en cada iteración del algoritmo. La propuesta está basada en una observación que realizamos al artículo en [3]. Notamos que en dicho artículo existen contradicciones y un error en el cálculo de la entropía minoritaria propuesta, por lo que hicimos la corrección correspondiente, y proponemos algunas otras mejoras como trabajo futuro.

Este artículo está organizado de la siguiente manera. Una revisión sobre trabajos relacionados es descrita en la sección 2. En la sección 3, se presentan los conceptos básicos relacionados con la medición del desempeño de métodos de clasificación con conjuntos de datos no balanceados. También se incluye una revisión breve sobre Weka, que es el framework utilizado para implementar la modificación propuesta. En la sección 4, se presenta el algoritmo original que se propuso en el artículo [3]. En esa sección mostramos la observación que realizamos, y un contra-ejemplo para demostrar un error en el cálculo de la entropía minoritaria. La corrección propuesta, así como la implementación, se detalla en la sección 5. Los resultados se presentan en la sección 6. Las conclusiones generales, así como trabajos futuros, se encuentran en la sección 7. En la última parte de este artículo se encuentran las referencias consultadas.

2. Trabajos relacionados

El efecto de desbalance en los datos sobre el desempeño de los clasificadores ha sido ampliamente estudiado durante los últimos años. Uno de los primeros trabajos sobre ello fue el descrito en [10], en el cual se usaron conjuntos de datos sintéticos para realizar una exploración sobre la relación que existe entre la complejidad de los conceptos subyacentes en los datos, el tamaño de los conjuntos de datos de entrenamiento y el nivel de desbalance entre las clases. Los métodos usados en [10] fueron árboles de decisión, redes neuronales y máquinas de soporte vectorial. En [5] y [11], se llevaron a cabo algunos experimentos aplicando el árbol de decisión C4.5 con conjuntos de datos reales. En [2] fue realizado un estudio de los algoritmos especialmente diseñados para la limpieza de los datos, y balance de los conjuntos de datos. Recientemente, en [9], fueron revisados los efectos del ruido en los datos sobre varios métodos de clasificación.

Algunas conclusiones sobre el problema de desbalance en los datos, que se han obtenido son las siguientes. El desbalance entre clases perjudica el desempeño de los métodos de clasificación, pero este no es el factor más importante [14]; el desbalance interno tiene un impacto más negativo que el desbalance entre

clases [7,10]; el ruido en los datos y los casos raros son perjudiciales para los clasificadores [9,13].

Probablemente, el algoritmo más famoso, diseñado específicamente para atender el desbalance de los conjuntos de datos es Synthetic Minority Oversampling Technique (SMOTE) [4]. Este sobre muestrea la clase minoritaria tomando cada muestra de la clase minoritaria e introduce nuevos objetos a lo largo de los segmentos de línea que unen a la instancia con los k vecinos más cercanos. Un problema con SMOTE es que tiende a sobre generalizar, y también introduce traslape entre clases, esto es por que las nuevas muestras son generadas sin considerar la distribución de los datos.

Una revisión más amplia sobre clasificación con conjuntos de datos no balanceados puede encontrarse en [8].

Uno de los algoritmos propuestos recientemente, y que utiliza un árbol de decisión es el presentado en [3]. En ese artículo, se propone usar tres particiones en la construcción del árbol. La idea es concentrar a las instancias de clase minoritaria en la partición central, usando el criterio de entropía minoritaria que se presenta más adelante en el presente artículo. De esta manera, el método le brinda mayor oportunidad a las instancias de clase minoritaria, de ser incluídas en nodos más puros. En la sección 4, se realiza una observación al algoritmo en [3], y se presenta una corrección del mismo.

3. Preliminares

3.1. Árbol de decisión C4.5, o J48 en Weka

En 1993, la Universidad de Waikato, en Nueva Zelanda, desarrolló la primera versión del software llamado *Weka* (*Waikato Environment for Knowledge Analysis*, que en español se traduce como “Entorno para análisis del conocimiento de la Universidad de Waikato”). Este software es un entorno de trabajo que provee la implementación de diferentes algoritmos para aprendizaje automático (*machine learning*). La finalidad principal fue hacer que los algoritmos de aprendizaje automático estén disponibles para el público en general, y que sean aplicados tanto en investigación, como en la industria. La versión actual de Weka es software libre publicado bajo la Licencia Pública General GNU, está escrito en lenguaje Java y soporta aplicaciones para *Big Data*.

Uno de los métodos para clasificación más populares en la comunidad de investigadores y practicantes del área de aprendizaje automático es C4.5 [12]. Este método es un árbol de decisión, ampliamente utilizado para realizar comparativas de desempeño de clasificadores en conjuntos de datos no balanceados. La implementación de C4.5 que ofrece Weka es llamada J48.

Los árboles de decisión son modelos cuya estructura tiene cierta similitud a los diagramas de flujo. La idea subyacente de este tipo de método es realizar particiones al conjunto de datos repetidamente, con la finalidad de que las particiones sean cada vez más *puras*, es decir, que contengan elementos de una sola clase. Cada partición o separación de un conjunto de datos (S), se realiza

probando todos los posibles valores de las instancias en cada dimensión o atributo (A), y después se selecciona a la mejor partición de acuerdo a algún criterio. Este proceso se realiza de manera recursiva.

Algunos de los criterios para elegir la mejor partición de conjuntos de datos, son basados en Entropía(E), Information Gain(IG), Split Info(SI) y Gain Ratio(GR), cuyas ecuaciones son mostradas a continuación :

$$E(S) = - \sum_{i=1}^N p_i \log(p_i) \quad (1) \quad SI(S) = - \sum_{i=1}^N \frac{S_i}{S} \left(\log\left(\frac{S_i}{S}\right) \right) \quad (3)$$

$$IG(S, A) = E(S) - \sum_{i=1}^N \frac{S_i}{S} E(S_i) \quad (2) \quad GR(S, A) = \frac{IG(S, A)}{SI(S)} \quad (4)$$

Los logaritmos son base dos para todas las fórmulas en este artículo.

3.2. Métricas para evaluar el desempeño de clasificadores en conjuntos de datos no balanceados

Una forma conveniente de observar el desempeño de un método de clasificación en conjuntos de datos con dos clases, es usar una matriz de confusión. Ésta contiene información sobre las predicciones hechas por el clasificador usando el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. La tabla 1 muestra un ejemplo de matriz de confusión.

Tabla 1. Matriz de confusión

Predicción		
↓		
positivo	negativo	
d	c	positivo
b	a	negativo
		↑
		Clase real

Utilizar únicamente la precisión de clasificación (ecuación (5)) para determinar el desempeño de un clasificador, no es conveniente cuando se trata de conjuntos de datos no balanceados. Esto es porque en este tipo de aplicaciones, se pueden obtener valores altos de precisión de clasificación, simplemente ignorando a la clase minoritaria, sin embargo, esta última es importante en casos como detección de transacciones fraudulentas, detección de intrusiones en la red, minería web, marketing directo, y diagnósticos médicos [4].

$$AC = \frac{a + d}{a + b + c + d} \quad (5)$$

Es común usar medidas tales como el recuerdo (*recall*) o true positive rate (TP), que indica la proporción de casos positivos que fueron correctamente identificados. El false positive rate (FP), que hace referencia a la proporción de elementos positivos que fueron clasificados incorrectamente como negativos. El true negative rate (TN), que indica la proporción de casos negativos que fueron clasificados correctamente. La proporción de elementos positivos que fueron clasificados incorrectamente como negativos, se obtiene con el false negative rate (FN). Es común que a la clase cuyo número de instancias es la menor (clase minoritaria), se le asigne la categoría de positivo, mientras que a la clase contraria (clase mayoritaria) se le asigne la de negativo. Las ecuaciones siguientes muestran la forma de calcular cada una de las medidas anteriores.

$$TN = \frac{a}{a + b} \quad (6)$$

$$TP = \frac{d}{c + d} \quad (9)$$

$$FP = \frac{b}{a + b} \quad (7)$$

$$FN = \frac{c}{c + d} \quad (8)$$

$$P = \frac{d}{b + d} \quad (10)$$

4. Árbol de decisión basado en entropía minoritaria

El árbol de decisión presentado en [3], divide el conjunto de datos en tres particiones, basándose en los conceptos de entropía minoritaria y tasa de ganancia. La idea es usar dos puntos de separación en el atributo seleccionado, de modo que permitan capturar la mayor cantidad de elementos de clase minoritaria en la partición central.

En [3], se propone la ecuación (11) para calcular la entropía minoritaria., en donde N representa la cantidad de particiones, n_i representa la cantidad de instancias de clase minoritaria en la i -ésima partición y n representa la cantidad de elementos de clase minoritaria.

$$MinorityEntropy(n) = - \sum_{i=1}^N \frac{n_i}{n} \left(\log \left(\frac{n_i}{n} \right) \right) \quad (11)$$

El propósito de usar esta medida, es dar mayor importancia a las instancias de clase minoritaria. Con la finalidad de reducir el costo computacional para la construcción del árbol de decisión, los puntos de separación se calculan usando un centroide c (ecuación (12)) con $y_i = +1$.

$$c = \frac{1}{N} \sum_{i=1}^N x_i \quad (12)$$

Este último es obtenido con los valores de instancias de clase minoritaria sobre el atributo bajo prueba.

En general, en cada etapa de la construcción del árbol, se calculan dos puntos de separación para cada atributo, que permiten dividir los datos en un nodo en tres particiones. Desde ahora, se denominarán SP_1 y SP_2 a los puntos de separación. El primero de estos es calculado desde la primera instancia de clase minoritaria del conjunto, hasta c , mientras que el segundo se calcula desde c , hasta la última instancia de clase minoritaria.

Todos los candidatos cuya tasa de ganancia sea mayor a un umbral T (ecuación 13), son almacenados y posteriormente se selecciona al candidato con el menor valor de entropía minoritaria.

$$T = 1 - R \quad (13)$$

En la ecuación 13, el valor de R , se calcula con la proporción de la tasa de varianza de las instancias minoritarias y la tasa de varianza de todas las instancias.

$$R = \frac{\sigma^2(\text{MinorityClassInstances})}{\sigma^2(\text{AllInstances})} \quad (14)$$

El árbol de decisión es construido de manera similar a C4.5, pero usando la medida de entropía mencionada anteriormente.

4.1. Una observación sobre el cálculo de la entropía minoritaria

La idea subyacente en el algoritmo presentado en [3], es concentrar la mayor cantidad de instancias de clase minoritaria en una sola partición durante la construcción de un árbol de decisión. Lo anterior es logrado mediante el uso del concepto de entropía minoritaria, explicado en la sección 3. Esta idea resulta atractiva para atacar el problema de clasificación sobre conjuntos de datos no balanceados, ya que se evita que el clasificador ignore a las instancias de clase minoritaria, tal como ocurre comúnmente con los algoritmos tradicionales.

Una observación que realizamos en el artículo [3], es que el valor de la entropía minoritaria no se encuentra acotado entre los valores cero y uno. En dicho artículo se puede encontrar la siguiente afirmación: “*This value lies between 0 and 1. If the minority entropy is 0, then all of minority class instances are located in the middle partition*”. Hemos encontrado que esto no es siempre cierto. Para mostrarlo, exponemos un contra ejemplo.

Suponer que tenemos un conjunto de datos con 100 elementos, de los cuales 10 son de clase minoritaria. Supongamos también que en alguna iteración del algoritmo, los puntos de separación SP_1 y SP_2 propician la formación de tres particiones con la distribución de los datos que se muestra en la figura 1.

Cada una de las particiones contiene la cantidad de datos mostrada en la tabla 2.

Al realizar el cálculo de la entropía minoritaria para cada partición, obtenemos lo siguiente:

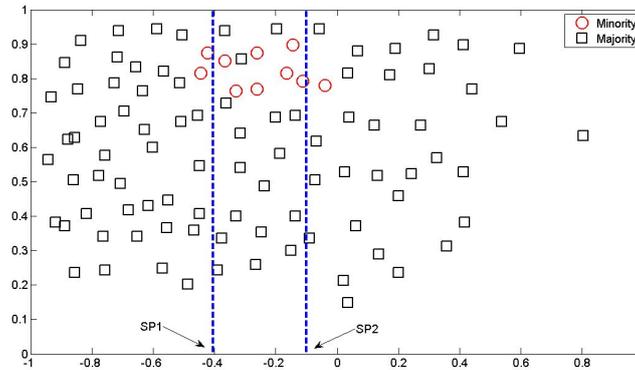


Fig. 1. Conjunto de datos sintético particionado

Tabla 2. Distribución de las instancias en las tres particiones de la figura 1

D	$Instancias_{maj}$	$Instancias_{min}$	Total
1	42	2	44
2	17	7	24
3	31	1	32

$$MinorityEntropy(n) = -\left(\frac{2}{10} \log\left(\frac{2}{10}\right)\right) - \left(\frac{7}{10} \log\left(\frac{7}{10}\right)\right) - \left(\frac{1}{10} \log\left(\frac{1}{10}\right)\right)$$

$$MinorityEntropy(n) \approx 1.16$$

Con lo obtenido en el contra ejemplo expuesto, es claro que la ecuación para calcular la entropía minoritaria produce valores que no se encuentran acotados entre cero y uno, como se afirma en [3]. Esto no sería un inconveniente, sino fuera porque al no estar acotada, cada posible valor en el atributo bajo prueba puede llegar a ser punto de separación, lo que hace que el algoritmo se comporte de manera similar a un árbol de decisión C4.5. En la siguiente sección, realizamos una propuesta para calcular de manera diferente la entropía minoritaria.

5. Método propuesto

Habiendo identificado el problema con el cálculo de la entropía minoritaria, se propone una modificación al algoritmo presentado en [3]. Descrito de manera

breve, se propone seguir manteniendo a las tres particiones para la construcción del árbol de decisión, sin embargo, para el cálculo de la entropía minoritaria se propone usar únicamente dos particiones, la idea es considerar a las particiones extremas como si se tratara de una sola.

El surgimiento de esta propuesta, está basado en observaciones que realizamos usando algunos conjuntos de datos sintéticos. Al analizar las condiciones para las cuales los valores de la entropía minoritaria pertenecían al intervalo $[0, 1]$, observamos que esto sólo ocurre cuando las instancias de clase minoritaria están presentes en dos de las tres particiones. En otros casos, el resultado obtenido puede estar fuera del rango indicado. La forma corregida para el cálculo de entropía minoritaria que proponemos en este artículo es la mostrada en la ecuación (15).

Es importante hacer notar, que pese a que nuestra corrección tiene cierta similitud a la fórmula de cálculo de entropía (ecuación (1)), entre ellas existe una gran diferencia. En la entropía se consideran todas las instancias de cada clase en el conjunto de datos de entrenamiento, en contraste, en la fórmula propuesta únicamente se consideran las instancias de clase minoritaria.

$$MinorityEntropy(n) = - \left(\frac{n_1}{n} \log \left(\frac{n_1}{n} \right) \right) - \left(\frac{n_2}{n} \log \left(\frac{n_2}{n} \right) \right) \quad (15)$$

Donde

n_1 representa la cantidad de instancias de clase minoritaria en la partición central,

n_2 representa la suma de la cantidad de instancias de clase minoritaria que se encuentran en las particiones izquierda y derecha, es decir, la suma de ambas,

n representa la cantidad de elementos de clase minoritaria antes de realizar particiones.

Para los casos en los que únicamente hay dos particiones, la partición central se considera la de la derecha. El pseudocódigo en Algoritmo 1 muestra en detalle la propuesta presentada.

Como ejemplo, suponer que los datos son los mismos del ejemplo anterior. Las instancias de la partición 1 han formado un sólo conjunto con los de la partición 2, por lo que el cálculo de la entropía minoritaria es ahora:

$$MinorityEntropy(n) = - \left(\frac{7}{10} \log \left(\frac{7}{10} \right) \right) - \left(\frac{3}{10} \log \left(\frac{3}{10} \right) \right) \approx 0.88129$$

La corrección del cálculo de entropía minoritaria ha sido probada tanto en casos triviales, así como en los casos en los que la fórmula original entrega resultados mayores a uno, aparentemente ambas formulas son similares.

En el Algoritmo 1, el intercambio de los atributos del conjunto L al conjunto R no debe afectar a este conjunto, ya que se realiza únicamente en el algoritmo, pero no en los datos, es decir; el recorrido que realiza el punto de separación SP_2 con cada atributo del conjunto R , no debe considerar los elementos del conjunto L que han sido agregados a R . Nuestra propuesta no asegura que en todos los caso las instancias de clase minoritaria se posicionen en la partición central.

Algorithm 1: C4.5 con Entropía minoritaria

Input : D : Conjunto de datos
Output: Árbol de decisión

begin

Crear un nodo del árbol;

if $\forall x \in MajorityClass$ **then**

| **return** *nodo como clase mayoritaria*

end

if $\forall x \in MinorityClass$ **then**

| **return** *nodo como clase minoritaria*

end

foreach *atributo* **do**

$c := \bar{D}_{min} : \{D_{min} : \exists x : x = instClaseMin\} \& D_{min} \subset D$;

$L := \{L : \exists x : x \leq c\}$;

$R := \{R : \exists x : x > c\}$;

foreach l *en* L **do**

foreach r *en* R **do**

| Calcular la tasa de ganancia para l y r ;

| Calcular la entropía minoritaria;

end

| Mover el atributo l al conjunto R ;

end

$G_{multiple} := \{\forall candidato : candidato > T\}$;

Seleccionar la menor entropía minoritaria de $G_{multiple}$;

end

Seleccionar el mejor candidato de los atributos como punto de separación;

Separar las instancias en las particiones D_1 , D_2 y D_3 correspondientes a los puntos de separación seleccionados;

Invocar recursivamente el algoritmo con D_1 , D_2 y D_3 ;

end

6. Resultados

Para probar el desempeño de nuestro algoritmo, se usaron los conjuntos de datos, cuyas características son mostradas en la tabla 3. Estos datos están públicamente disponibles en la Internet [1], y son usados para realizar experimentos con conjuntos de datos no balanceados. En la versión actual de la implementación, el tipo de atributo considerado por el algoritmo es únicamente numérico, es decir, no se soportan atributos nominales, ni tipo cadena, etc.

El código fuente del algoritmo propuesto, fue escrito en lenguaje Java, modificando varias clases que componen al algoritmo J48 de Weka.

Los datos reportados, son el promedio de 30 repeticiones de los experimentos para cada conjunto de datos. En cada experimento, se usaron el 70 % de los datos elegidos pseudo aleatoriamente para entrenamiento, y el 30 % restante para prueba.

Tabla 3. Conjuntos de datos usados en los experimentos

Nombre	Mayoritaria	Minoritaria	Atributos	Total	IR
ecoli4	316	20	8	336	15.80
glass-0-1-6_vs_5	175	9	10	184	19.44
page-blocks-1-3_vs_4	444	28	11	472	15.85
pima	500	268	9	768	1.86
shuttle-c0-vs-c4	1,706	123	10	1,829	13.86
vehicle1	629	217	19	846	2.89
vowel0	898	90	14	988	9.97
yeast-2_vs_4	463	51	9	514	9.07

La tabla 4 muestra los resultados obtenidos. El significado de las columnas está descrito en la sección 3.

Es interesante notar que aunque a lo largo del artículo [3], se explica un método que pareciera trabajar con atributos numéricos, en los resultados de ese trabajo, los autores presentan solamente conjuntos de datos con atributos categóricos. Debido a lo anterior, y a que la implementación actual de nuestro algoritmo sólo soporta atributos numéricos, fue imposible realizar una comparativa.

Como puede observarse, el algoritmo propuesto tiene un desempeño que supera al del C4.5 original. Es necesario que nuestra propuesta soporte otros tipos de atributos, y que se realicen más experimentos mostrando resultados como áreas bajo la curva (ROC), para poder llegar a conclusiones más generales.

7. Conclusiones y trabajo futuro

El problema de clasificación con conjuntos de datos no balanceados, ha sido un área de interés para la comunidad tanto teórica como práctica de minería de datos. Se han realizado una cantidad grande de estudios para tratar de entender este problema complejo, y también se han propuesto una amplia plétora de métodos para atacar la dificultad de obtener un buen desempeño de los algoritmos de clasificación en este tipo de conjuntos de datos.

En este artículo, se propone una corrección a un método publicado recientemente en [3], está basado en un árbol de decisión y en el concepto de entropía minoritaria. El error en dicha publicación fue encontrado al tratar de implementar el método, y se observaron las condiciones para las cuales sucede una contradicción en la definición de la entropía minoritaria. Al momento de la publicación de este artículo, el método propuesto solamente soporta conjuntos de datos con atributos numéricos, y se encuentra implementado en lenguaje de programación Java, además, continúa nuestra investigación con el objetivo de realizar una comparativa más justa entre el algoritmo de [3] y el propuesto, e incluir más resultados con métricas como F-measure, AUC ROC y PCR.

Los trabajos futuros de esta investigación son los siguientes:

Tabla 4. Resumen de resultados

Método	Recall	FP	TN	FN	Precisión
yeast-2_vs_4					
Propuesta	0.732	0.035	0.965	0.268	0.685
C4.5	0.771	0.040	0.960	0.229	0.636
vowel0					
Propuesta	0.921	0.012	0.988	0.079	0.882
C4.5	0.935	0.014	0.986	0.065	0.857
vehicle1					
Propuesta	0.530	0.176	0.824	0.470	0.456
C4.5	0.536	0.179	0.821	0.464	0.443
shuttle-c0-vs-c4					
Propuesta	0.988	0.000	1.000	0.012	0.998
C4.5	0.989	0.000	1.000	0.011	1.000
pima					
Propuesta	0.628	0.218	0.782	0.372	0.578
C4.5	0.643	0.222	0.778	0.357	0.560
page-blocks-1-3_vs_4					
Propuesta	0.948	0.008	0.992	0.052	0.884
C4.5	0.915	0.005	0.995	0.085	0.925
glass-0-1-6_vs_5					
Propuesta	0.426	0.040	0.960	0.574	0.238
C4.5	0.595	0.024	0.976	0.405	0.560
ecoli4					
Propuesta	0.826	0.020	0.980	0.174	0.672
C4.5	0.779	0.022	0.978	0.221	0.638

- Realizar una implementación paralela, distribuida o en GPU del método propuesto, para aplicaciones con datos grandes.
- Mejorar el método para poder procesar más tipos de atributos.
- Realizar una comparativa con otros métodos.
- Proponer otras reglas para particionar datos durante la construcción del árbol de decisión.
- Reducir el tiempo de entrenamiento.
- Realizar híbridos con métodos externos para mejorar el desempeño del clasificador.
- Realizar más pruebas y analizar profundamente los resultados obtenidos.

Agradecimientos. Los autores de este artículo agradecen a los revisores anónimos por sus valiosas observaciones para mejorar este trabajo. También, agradecen a la Universidad Autónoma del Estado de México, por el apoyo económico a través del proyecto de investigación 3790/2014/CID.

Referencias

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S.: KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing* 17(2-3), 255–287 (2011), <http://www.oldcitypublishing.com/MVLSC/MVLSAbstracts/MVLS17.2-3abstracts/MVLSv17n2-3p255-287Alcala.html>
2. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.* 6(1), 20–29 (Jun 2004), <http://doi.acm.org/10.1145/1007730.1007735>
3. Boonchuay, K., Sinapiromsaran, K., Lursinsap, C.: Minority split and gain ratio for a class imbalance. In: *Fuzzy Systems and Knowledge Discovery (FSKD)*, 2011 Eighth International Conference on. vol. 3, pp. 2060–2064 (2011)
4. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002), <http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a.pdf>
5. Drummond, C., Holte, R.C.: C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. pp. 1–8 (2003)
6. Elkan, C.: The foundations of cost-sensitive learning. In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*. pp. 973–978. IJCAI'01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
7. Gong, W., Luo, H., Fan, J.: Extracting informative images from web news pages via imbalanced classification. In: *Proceedings of the 17th ACM International Conference on Multimedia*. pp. 1123–1124. MM '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1631272.1631529>
8. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Trans. on Knowl. and Data Eng.* 21(9), 1263–1284 (Sep 2009), <http://dx.doi.org/10.1109/TKDE.2008.239>
9. Hulse, J.V., Khoshgoftaar, T.: Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering* 68(12), 1513 – 1542 (2009), <http://www.sciencedirect.com/science/article/pii/S0169023X09001141>
10. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intelligent Data Analysis* 6(5), 429 (2002)
11. Luengo, J., Fernández, A., Herrera, F., Herrera, F.: Addressing data-complexity for imbalanced data-sets: A preliminary study on the use of preprocessing for c4.5. In: *Intelligent Systems Design and Applications, 2009. ISDA '09. Ninth International Conference on*. pp. 523–528 (2009)
12. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
13. Seiffert, C., Khoshgoftaar, T., Van Hulse, J.: Hybrid sampling for imbalanced data. In: *Information Reuse and Integration, 2008. IRI 2008. IEEE International Conference on*. pp. 202–207 (2008)
14. Weiss, G.M.: Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl.* 6(1), 7–19 (Jun 2004), <http://doi.acm.org/10.1145/1007730.1007734>

Segmentación del padrón de sujetos vulnerables en el Sistema de Asistencia Alimentaria

Juan Pablo Granados García¹, Rosario Baltazar¹, Silvia Quintana Vargas²,
Claudia Leticia Díaz González¹, Martha Alicia Rocha Sánchez¹, Arturo
Hernández Aguirre³

¹ División de Estudios de Posgrado e Investigación,
Instituto Tecnológico de León,
León, Guanajuato, México

² Hospital Regional de Alta Especialidad del Bajío,
León, Guanajuato, México

³ Centro de Investigaciones en Matemáticas,
Departamento de Ciencias de la Computación,
Guanajuato, México

<http://posgrado.itleon.edu.mx>
artha@cimat.mx

Resumen. En este trabajo se llevó a cabo una caracterización de sujetos vulnerables en el Sistema de Asistencia Alimentaria del DIF del Estado de Guanajuato, esto aplicando algoritmos de agrupamiento. La segmentación se realizó con la información del Padrón 2013-2014. Se aplicó un proceso de limpieza y normalización de los datos. Además se creó un vector de características, al cual se le aplicaron algoritmos de agrupamiento como el Método de Ward y el K-means. Posteriormente se contrastó con algoritmos de clasificación de mínima distancia, KNN, Red Neuronal Artificial Back-propagation y Red Neuronal Artificial con PSO.

Palabras clave: minería de datos, segmentación, agrupamiento, clasificación.

1. Introducción

Los programas de ayuda alimentaria en México tienen como objetivo contribuir en la canasta básica de los beneficiarios para mejorar su nutrición. Los programas de alimentos y cocinas del Sistema Nacional para el Desarrollo Integral de la Familia (DIF) y de las Organizaciones No Gubernamentales (ONG) tienen una cobertura muy baja, por lo tanto, es importante hacer una distribución eficiente de los recursos para lograr beneficiar a los sujetos más vulnerables[23].

Una forma para lograr una distribución eficiente de los recursos es utilizar reconocimiento de patrones para generar grupos de beneficiarios con características similares. En el 2000 [25] se elabora un trabajo que resume y compara algunos de los métodos conocidos y utilizados en diversas etapas de un sistema de

reconocimiento de patrones. Además demuestra que el diseño de un sistema de reconocimiento requiere varias etapas: definición de clases, el medio de detección, la representación de patrones, la extracción de características y selección, el análisis y agrupamiento, el diseño del clasificador y el aprendizaje, la selección de muestras de entrenamiento y finalmente la evaluación del desempeño.

Posteriormente en el 2010 [27] se presenta un nuevo método para analizar la relación entre las probabilidades producidas por un modelo de clasificación y la variación de sus valores de entrada. El objetivo es aumentar la probabilidad de predicción de una clase dada. Mediante la exploración de los posibles valores de las variables de entrada.

Ahora bien la Escala Latinoamericana y Caribeña de Seguridad Alimentaria (ELCSA) presenta su validez interna y externa en 2010 [24]. Se recomienda su aplicación por la Secretaría de Seguridad Alimentaria y Nutricional (SESAN) y la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO). ELCSA responde a la necesidad de ampliar y mejorar la medición del hambre, usando métodos para medir directamente la experiencia en los hogares ante la inseguridad alimentaria y hambre.

En el 2013 [10] se realizó una investigación con respecto a la asistencia alimentaria en México [10] que utiliza nuevas tecnologías que favorezcan o apoyen el análisis, resultados e implementaciones. Bajo el contexto anterior, se plantean modelos matemáticos que permitan el uso de algoritmos de clasificación y reconocimiento de los diferentes modelos de seguridad alimentaria.

En este trabajo de investigación se delimita en lo particular al Sistema de Asistencia Alimentaria del organismo DIF del Estado de Guanajuato. Una de las tareas del DIF es la identificación de la inseguridad alimentaria y hace hincapié en la reducción del hambre y la desnutrición en las familias con bajos recursos.

Con esta investigación se pretende generar grupos con características propias que nos permite conocer el comportamiento y la tendencia de los sujetos vulnerables, utilizando técnicas de minería de datos y reconocimiento de patrones. Por lo tanto, se convierten los datos en conocimiento útil. Se aporta nueva información de apoyo a la labor de identificación, para mejorar la distribución de los recursos y un control efectivo de los beneficiarios.

De esta manera es posible crear un perfil particular con las diferentes tendencias en base a los indicadores sociales de los sujetos y poder atender en su momento a ciertas áreas que requieran de un determinado apoyo más que otras.

2. Marco teórico

2.1. Agrupación de datos

Agrupación de datos o clasificación no supervisada es un proceso de asignación de un conjunto de registros en subconjuntos, llamados Clusters, de tal manera que los registros en el mismo grupo son similares y registros en diferentes grupos son muy diferentes [7,2].

2.2. Clasificadores no supervisados

Método de Ward: Es un algoritmo de agrupamiento jerárquico que divide un conjunto de datos en una secuencia de particiones anidadas representados por un diagrama de árbol o dendrograma [5,6]:

Paso 1: un nuevo cluster C_n está formado por la fusión de dos grupos en el conjunto inicial de clusters $F_0 = \{C_0, C_1, \dots, C_{n-1}\}$. Luego, después de la etapa 1 y antes del paso 2, se tiene un conjunto de clusters $F_1 = \tilde{F}_0 \cup \{C_n\}$ donde \tilde{F}_0 es serie de clusters sin combinar en F_0 . Si C_0 y C_1 tienen la distancia mínima entre todos los pares de grupos, entonces $C_n = C_0 \cup C_1$ y $\tilde{F}_0 = \{C_2, C_3, \dots, C_{n-1}\} = F_0 \setminus \{C_0, C_1\}$.

Paso 2: un nuevo grupo C_{n+1} está formado por la fusión de dos grupos en el conjunto de clusters F_1 . Del mismo modo, dejamos \tilde{F}_1 el conjunto de agrupaciones sin combinar en F_1 . Luego después del paso 2 y antes del paso 3, tenemos un conjunto de clusters $F_2 = \tilde{F}_1 \cup \{C_{n+1}\}$. El algoritmo continúa este proceso hasta que en el paso $n - 1$ cuando los dos últimos grupos se fusionaron para formar el cluster C_{2n+2} . Después del paso $n - 1$, tenemos $F_{n-1} = \{C_{2n+2}\}$, que contiene un solo cluster. El algoritmo se detiene después de la etapa $n - 1$.

En el proceso anterior, se tiene $|F_0| = n$, $|F_1| = n - 1, \dots, |F_{n-1}| = 1$, donde $|\cdot|$ denota el número de elementos en el conjunto. Para decidir que grupos se fusionara, se tiene que calcular las distancias entre las agrupaciones. Para calcular la distancia entre un de cluster y un nuevo grupo formado por dos grupos Lance-Williams propone [30].

Antes del paso i ($1 \leq i < n - 1$), se tiene un conjunto de clusters F_{i-1} , que contiene $n - i + 1$ clusters. Suponiendo que C_{i_1} y C_{i_2} tienen la menor distancia entre todos los pares de clusters en F_{i-1} . Entonces C_{i_1} y C_{i_2} se fusionarán para formar el cluster C_{n+i+1} . Para calcular la distancia entre un cluster $C \in \tilde{F}_{i-1} = F_{i-1} \setminus \{C_{i_1}, C_{i_2}\}$ es:

$$D(C_{i_1}, C_{i_2}) = \sqrt{\frac{2|C_{i_1}||C_{i_2}|}{|C_{i_1}| + |C_{i_2}|}} d(c_{i_1}, c_{i_2}) \quad (1)$$

donde c_{i_1} y c_{i_2} representan los centroides de los clusters C_{i_1} y C_{i_2} , cuya distancia se está midiendo, $d(c_{i_1}, c_{i_2})$ es una métrica que devuelve la distancia entre ambos centroides.

K-means: El algoritmo divide el conjunto de datos en k grupos C_0, C_1, \dots, C_{n-1} , reduciendo la función objetivo 2 [5,2]:

$$J = \sum_{j=1}^n \sum_{k=1}^K u_{kj} * \left[(\bar{x}_j - \bar{z}_k)^\lambda \right]^{1/\lambda} \quad (2)$$

Donde u_{kj} es igual a 1 si el j -ésimo punto pertenece al grupo k y 0 en caso contrario, \bar{z}_k denota el centro del grupo k , \bar{x}_j denota la j -ésimo punto de los datos. Los diferentes pasos del algoritmo K-means son:

Paso 1: Seleccionar K centroides de los grupos z_1, z_2, \dots, z_k de manera aleatoria para los n puntos x_1, x_2, \dots, x_n .

Paso 2: Asignar el punto $x_i, i = 1, 2, \dots, n$ al cluster $C_j, j \in 1, 2, \dots, k$ si solo si $\left[(\bar{x}_i - \bar{z}_j)^\lambda \right]^{1/\lambda} < \left[(\bar{x}_i - \bar{z}_p)^\lambda \right]^{1/\lambda}, p = 1, 2, \dots, K, j \neq p$. Los empates se resuelven de manera arbitraria.

Paso 3: Calcular nuevo centroides de los grupos $z_1^*, z_2^*, \dots, z_k^*$ como sigue: $z_i^* = \frac{\sum_{x_j \in C_i} x_j}{n_i}, i = 1, 2, \dots, K$, donde n_i es el número de elementos que pertenecen al grupo C_j .

Paso 4: El proceso de partición se repite hasta que se cumpla una sentencia de paro: a) Los centros de los grupos no cambian, b) El valor J se hace menor que un umbral, c) El número máximo de iteraciones se han agotado. De lo contrario $z_i = z_i^*, i = 1, 2, \dots, K$ y continuar desde el paso 2.

2.3. Reconocimiento de patrones

El reconocimiento de patrones es la disciplina científica cuyo objetivo es la clasificación de objetos en cierto número de categorías o clases. Dichos objetos pueden ser imágenes, señales o cualquier medida que pueda ser clasificada, los cuales se denominan patrones [8]. En el campo de reconocimiento de patrones se refiere a la detección automática de regularidades en los datos mediante el uso de algoritmos de computadora y con el uso de estas regularidades poder tomar acciones tales como la clasificación de los datos en diferentes categorías [9].

2.4. Clasificadores supervisados

Clasificadores de distancia mínima: Es el más utilizado para la clasificación de patrones de diferente tipo. Parte del hecho que las clases son linealmente separables. Se supone que la pertenencia de cada patrón se conoce y además se sabe el número de clases en las que el patrón puede ser clasificado [11,12]. Los vectores de características son asignados a clases de acuerdo a su distancia entre los puntos medios respectivos.

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma^{-1} (x - \mu_i) \quad (3)$$

Clasificadores de tipo k-vecinos KNN: Considerando un conjunto de N puntos, $x_1, x_2, \dots, x_N \in R^l$ que provienen de una distribución estadística desconocida. El objetivo es estimar el valor de la función de densidad de probabilidad desconocida en un punto dado x . De acuerdo con la técnica de estimación vecino k-más cercano, se realizan los siguientes pasos [8]:

1. Elegir un valor de k .
2. Encuentra la distancia entre x y todos los puntos de entrenamiento $x_i, i = 1, 2, \dots, N$.
3. Encontrar los puntos k-más cercanos a x .
4. Calcule el volumen $V(x)$ en el que los k-vecinos más cercanos.
5. Calcular la estimación por $p(x) \approx \frac{k}{NV(x)}$

Clasificadores Neuronales: Una red neuronal artificiales (RNA) es un procesador masivamente paralelo distribuido que es propenso por naturaleza a almacenar conocimiento experimental y hacerlo disponible para su uso [11]. Este mecanismo de conocimiento se parece al cerebro: *a)* Es adquirido a través del aprendizaje, *b)* Se almacena mediante la modificación del peso sináptico de las distintas uniones entre neuronas. Una RNA pueden ser usados para la clasificar patrones de clases lineal o no linealmente separables.

Back-propagation (BP) es un método de aprendizaje supervisado de redes neuronales que consta de una capa de neuronas de entrada, otra de neuronas de salida y opcionalmente una o varias capas de neuronas ocultas [29]. Tiene la forma de propagar hacia atrás el error, desde la capa de salida a la de entrada, modificando los pesos de las capas intermedias. Permitiendo así aprender mediante un conjunto de ejemplo y obteniendo una salida coherente para una entrada de salida.

Particle Swarm Optimization (PSO) es un método metaheurístico proporcional basado en poblaciones. Se basa libremente en el comportamiento de un grupo coordinado, como el de congregación de aves. Mantiene una colección de partículas virtuales donde cada partícula representa una potencial mejora de la solución a un problema, que en el caso de las RNA, es un conjunto de valores para los pesos y sesgos que reduzcan al mínimo el error entre los valores de salida calculados y valores de salida conocidos en un conjunto de datos de entrenamiento [28].

2.5. Indicadores sociales

Nivel Socioeconómico (NSE): Es la norma creada por la Asociación Mexicana de Inteligencia de Mercado y Opinión Pública (AMAI) [13,18], basada en análisis estadístico, que permite agrupar y clasificar a los hogares mexicanos en siete niveles, de acuerdo a su capacidad para satisfacer las necesidades de sus integrantes en términos de: vivienda, salud, energía, tecnología, prevención y desarrollo intelectual.

$$NSE = \sum_{i=1}^{10} V_i \quad (4)$$

Donde $V_{i=1}$ Televisiones a color, $V_{i=2}$ Computadoras, $V_{i=3}$ Número de focos, $V_{i=4}$ Número de autos, $V_{i=5}$ Estufa, $V_{i=6}$ Baños Completos, $V_{i=7}$ Regadera, $V_{i=8}$ Tipo de piso, $V_{i=9}$ Número de habitaciones, $V_{i=10}$ Educación del jefe de familia.

Nivel Vulnerabilidad (NV): Se define desde tres dimensiones críticas[14]: *a)* Como un efecto directo/resultado, *b)* Resultado de varios factores de riesgo, *c)* La incapacidad de manejar tales riesgos.

$$NV = \sum_{i=1}^{12} V_i \quad (5)$$

Donde $-V_{i=1}$ Contar discapacidad igual a 7, $-V_{i=2}$ Contar discapacidad igual a 6, $-V_{i=3}$ Contar discapacidad igual 999, $V_{i=4}$ Contar número de discapacidades, $V_{i=5}$ Enfermedad Grave, $V_{i=6}$ Mujer embarazada, $V_{i=7}$ Mujer lactante, $V_{i=8}$ Alcohol, $V_{i=9}$ Drogas, $V_{i=10}$ Menor de 5, $V_{i=11}$ Mayor de 65, $V_{i=12}$ Migrante.

Nivel Discapacidad (ND): Es la adjetivación de la deficiencia en el sujeto y con una repercusión directa en su capacidad de realizar actividades en los términos considerados normales para cualquier sujeto de sus características [15].

$$ND = -V1 - V2 \quad (6)$$

Donde $V1 =$ Contar discapacidad igual 7, $V2 =$ Contar discapacidad igual 6.

Nivel Seguridad Alimentaria (NSA): Se define como la garantía de que los individuos, las familias y la comunidad en su conjunto, accedan en todo momento a suficientes alimentos inocuos y nutritivos, principalmente producidos en el país en condiciones de competitividad, sostenibilidad y equidad, para que su consumo y utilización biológica les procure óptima nutrición, una vida sana y socialmente productiva, con respeto de la diversidad cultural y preferencias de los consumidores [16,18].

$$NSE = \sum_{i=1}^{15} V_i \quad (7)$$

Hacinamiento (HC): Número de integrantes de la familia entre número de cuartos, se considera hacinamiento valores superiores a 2 [18].

$$HC = \frac{Integrantes}{Cuartos} \quad (8)$$

Problemas de Salud (PS): La salud no es sólo la ausencia de afecciones o enfermedades, sino también es el estado de bienestar somático, psicológico y social del individuo y de la colectividad [21,20].

$$PS = V1 - V2 - V3 \quad (9)$$

Donde $V1 =$ Contar el número de problemas de salud, $V2 =$ Contar igual a 11, $V3 =$ Contar igual a 10.

Más Vulnerable (MV): El DIF maneja la ecuación 10 para la obtención de un parámetro numérico que resuma al sujeto a ser seleccionado.

$$MV = \begin{cases} NSA + (388 - NSE) + HC + NV + ND + PS, m = 0 \\ NSA + (388 - NSE) + HC + NV + ND + PS - 28, m = 1 \end{cases} \quad (10)$$

Índice de Necesidades Básicas Insatisfechas (INBI): Bajo esta concepción de pobreza, la CEPAL (Comisión Económica para América Latina y el Caribe) diseñó el método de medición de las Necesidades Básicas Insatisfechas (NBI) [17], para clasificar los hogares como pobres y no pobres. Las personas que pertenecen a un hogar con una necesidad insatisfecha se consideran como pobres $INBI = 1$ y con $INBI > 1$ se califican en una situación de miseria o pobreza extrema.

$$INBI = VI + VS + HC + DE + IE \quad (11)$$

Donde VI = vivienda inadecuada que presenta piso de tierra, VS = vivienda sin servicios considerada sin agua por acueducto o sin conexión a alcantarillado o a pozo séptico, HC = hacinamiento, DE = dependencia económica de los hogares cuyo jefe tenga un nivel educativo inferior a tercero de primaria y tres o más personas por cada persona ocupada, IE = inasistencia escolar a los hogares en los cuales algún niño entre 7 y 11 años, pariente del jefe, no asista a algún establecimiento educativo.

Canasta Básica: Conjunto de alimentos cuyo valor sirve para construir la línea de bienestar mínimo [19]. Éstos se determinan de acuerdo con el patrón de consumo de un grupo de personas que satisfacen con ellos sus requerimientos de energía y nutrientes.

3. Resultados actuales

Para cada una de las técnicas y algoritmos utilizados se programaron en C# con C++: a) C# es la parte visual e interactiva que inicializa cada algoritmo y presenta los resultados generados, b) C++ llegan los parámetros necesarios para cada algoritmo, realiza cada uno de los cálculos y regresa los resultados.

Inicialmente, los datos usados para la realización de este estudio, correspondieron al Padrón 2013-2014 del DIF, que pertenecen a los posibles sujetos vulnerables registrados en el estado de Guanajuato. El Padrón 2013-2014 con tiene la información recolectada con el instrumento de identificación de sujetos vulnerables basado en ELCSA. Entonces se tienen un conjunto inicial de 8397 familias con 27255 registros de posibles sujetos vulnerables que conforman todas las familias.

Se hizo un preprocesamiento de los datos. En esta etapa, se realizó la limpieza, eliminando los datos fuera de rango, datos atípicos y aquellos considerados inconsistentes. Se encontraron 232 datos con una escala de seguridad alimentaria inconsistente que se salía de los estándares empleados por el ELCSA. Por tanto, se utilizó un método de estimación, el cual consiste en eliminar dichos registros, quedando entonces 27023.

Además la presencia de datos faltantes en diferentes registros, se optó el algoritmo Non-linear Iterative Projections by Alternating Least-Squares (NIPALS) como método para estimar los datos faltantes. El algoritmo NIPALS obtiene

estimaciones de las componentes y de los vectores que permiten describir la matriz de datos y estimar los datos faltantes [3,4]. El algoritmo extrae una componente principal a la vez.

Se procedió a seleccionar las variables que mejor representen la variabilidad de los datos. Por lo tanto, se realizó un análisis con estadísticos descriptivos para la selección características con una varianza diferente de cero. Estas correspondieron al nivel escolaridad, tipo de ocupación, ingreso del jefe de familia, problemas de salud, tipo de discapacidad, tipo de vulnerabilidad, servicio médico, elementos de vivienda, violencia, integrantes en la familia, presencia de menores, gasto alimentario, seguridad alimentaria y canasta alimentaria.

Considerando que las variables pueden manejar diferentes escalas, que puede implicar a que aquellas con un mayor rango de valores les quiten importancia a otras con un menor rango, todas las variables consideradas fueron normalizadas. Se estandarizo los datos con respecto a la normalización Min-Max [1], es un enfoque simple donde se fijan los valores mínimos y máximos de las variables normalizadas entre $[0, 1]$.

Dada la gran dimensionalidad del problema por la cantidad de características, se hizo indispensable la reducción de estas, para posibilitar y hacer más eficiente el análisis. Se aplicó un Análisis de Componentes Principales (ACP) [4] con el cálculo de los valores propios de la matriz de correlaciones y generando un total de 25 componentes con una descripción de 89.334 %.

La medida de métrica de distancia seleccionada para la aplicación de los diferentes algoritmos de agrupamiento y clasificación fue la Minkowski de orden $\lambda = 3$. Esta es una medida general distancia [7] comúnmente utilizada.

3.1. Primer experimento

Se aplicó el algoritmo Método de Ward, se produjo el dendrograma de la figura 1. Seleccionar un dendrograma en un nivel particular produce una parti-

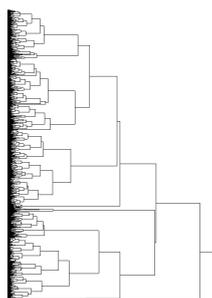


Fig. 1. Dendrograma generado por el método de Ward.

ción en grupos g disjuntos. Por lo tanto el dendrograma tiene un máximo 27023 grupos distintos. Para el análisis y busca del comportamiento de las diferentes

tendencias y definición de los grupos, se fue descendiendo por los niveles del dendrograma.

Se analizó cada uno de los grupos generados al descender por cada nivel, obteniendo de este modo las tendencias en los grupos mostrados en la tabla 2. Se generaran grupos explicativos del comportamiento y tendencias que ayudan a comprender el comportamiento de los datos.

Por otra parte, se utilizó el K-means para visualizar si persisten tales tendencias con los siguientes parámetros: grupos 18, iteraciones 100 y umbral 1. Se presentó una similitud ± 50 , tabla 1 y se validó el modelo de comportamiento.

Tabla 1. 18 grupos y sus tamaños.

Cluster	Núm. Personas	Cluster	Núm. Personas
1	3704	10	379
2	3248	11	181
3	1577	12	132
4	3043	13	2442
5	1187	14	2174
6	40	15	2034
7	2353	16	1436
8	9	17	190
9	2776	18	118

Los grupos están en función de los diferentes indicadores sociales: violencia, vivienda inadecuada y sin servicios, dependencia económica, inasistencia escolar, INBI y canasta básica. Así también con NSE, NV, ND, NSA, HC y PS usados por el DIF para obtener a los sujetos MV.

Cluster 1 Presencia de viviendas inadecuadas pero con servicios básicos y con hacinamiento. No existe la dependencia económica e inasistencia escolar. Tienen un INBI de pobre y pobre extremo, compuesta casi en su totalidad por NSE E, poca presencia NV, PS y nula por ND. Además la canasta básica con el gasto alimentario es muy parecidos.

Cluster 2 Presencia de familias grandes compuestas por 10 integrantes y con 2 menores en promedio por familia. Viven en viviendas inadecuadas, sin servicios básicos y con hacinamiento. Existe la dependencia económica, tiene un INBI como pobres extremos, compuesta casi en su totalidad por NSE E, presencia de NV por el número de menores, sin ND y algunos PS.

Cluster 3 Presencia de familias grandes compuestas por 9 integrantes y con 2 menores en promedio por familia. Viven en viviendas inadecuadas, sin servicios básicos y con hacinamiento. Existe la dependencia económica, tiene un INBI como pobres extremos, compuesta casi en su totalidad por NSE D, presencia de NV por el número de menores, sin ND y algunos PS.

Cluster 4 Presencia de viviendas inadecuadas sin servicios básicos y hacinamiento. No existe la dependencia económica e inasistencia escolar. Tienen

Tabla 2. Clusters generados.

Cluster	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
Vivienda	Núm. Personas	3745	3298	1530	3044	1145	40	2335	9	2806	379	181	132	2440	2197	2098	1336	190	118	
	Inadecuada	3745	3298	1530	3032	1130	28	4	0	96	309	157	96	2432	2197	2083	1311	142	93	
	Sin Servicios	31	3292	1528	3040	1104	31	2087	0	72	273	138	101	1520	2197	64	0	7	114	85
	Dependencia Económica	9	3262	1521	4	20	8	31	0	2746	133	50	29	40	0	0	0	0	44	26
	Inasistencia Escolar	0	0	0	0	0	0	0	0	0	379	0	1	0	0	0	0	0	4	6
	Pobre	1283	7	0	3	2	9	180	0	2794	0	16	15	655	0	2069	1309	24	26	26
	Pobre Extremo	2462	3281	1530	3041	1143	31	2155	0	12	379	165	117	1785	2197	29	2	166	92	92
	Núm. integrantes	5	10	9	5	4	4	4	3	5	5	4	4	3	2	2	3	5	4	4
	Núm. menores	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
D	243	182	1498	75	263	17	1905	0	611	96	36	25	568	0	231	852	55	64	64	
NSE	E	3502	3116	32	2947	882	23	393	0	2066	283	145	107	1872	2197	1867	6	135	54	
Si	3477	3232	1505	2297	867	18	68	0	580	291	90	48	2150	60	86	302	104	55	55	
Hacinamiento	No	268	66	25	347	278	22	2267	9	2226	88	91	84	290	2137	2012	1034	86	63	
0	3685	2146	927	3418	26	24	2120	3	1726	379	117	70	994	1069	1058	635	130	90	90	
NV	1	153	1068	603	6	1101	16	214	3	953	0	63	60	1444	1128	1040	697	60	28	
2	7	14	3	0	18	0	1	0	127	0	1	2	164	0	0	4	0	0	0	
0	3745	3207	1530	3041	1142	34	2335	4	2806	350	165	116	1	2197	2098	1324	155	93	93	
1	0	91	0	3	2	3	0	0	0	21	15	14	2034	0	0	12	27	21	21	
2	0	0	0	0	1	3	0	0	0	7	1	2	377	0	0	0	8	3	3	
3	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	1	0	0	5	0	0	0	0	0	0	
ND	0	0	0	0	0	0	0	0	0	0	0	103	1013	1427	1425	2	131	81	81	
1	3567	3220	1338	2988	1117	23	2273	8	2684	349	145	103	1124	769	670	802	48	26	26	
2	169	78	192	5	26	14	60	0	115	30	32	19	1124	269	1	3	473	9	10	
3	9	0	4	3	2	3	2	0	7	0	4	10	34	0	0	53	2	1	1	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
PS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
En Hogar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Que hizo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Fallecimiento	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Desaparición	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Violencia	0	0	0	20	0	0	0	0	0	0	152	132	0	0	0	0	0	0	18	
Gasto Alimentación	355	435	438	413	392	210	468	8500	355	388	439	340	261	228	225	300	396	328	328	
Canasta Básica	355.63	357.77	359.03	354.36	354.97	358.39	594.4	319.25	357.93	351.86	357.82	355.48	362.37	371.72	367.22	343.93	348.99	334.60	334.60	
Máximo	421	419	419	424	419	415	412	320	428	426	422.5	421.3	430	423	423	428	413	399.5	399.5	
Mínimo	284	290	292	289.7	297	319	298.4	217.5	271	264.3	297	293	254.3	326	289.5	219.5	259.7	267.2	267.2	
MV por DIF																				

un INBI de pobre extremo, está compuesta casi en su totalidad por NSE E y no presentan NV, ND y PS.

Cluster 5 Presencia de viviendas inadecuadas sin servicios básicos y hacinamiento. No existe la dependencia económica e inasistencia escolar. Tienen un INBI de pobre extremo, está compuesta por NSE D y E, presentan NV, pero no ND y PS.

Cluster 6 Presencia de sujetos registrados pero que están fallecidos en su totalidad.

Cluster 7 Presencia de viviendas adecuadas pero sin servicios básicos y sin hacinamiento. No existe la dependencia económica e inasistencia escolar. Tienen un INBI de pobre extremo, está compuesta casi en su totalidad por NSE D, poca presencia NV, ND y PS y la canasta básica es la más alta.

Cluster 8 Sujetos con NSE C y un gasto alimentario alto.

Cluster 9 Presencia de viviendas adecuadas, con servicios básicos y con poco hacinamiento. Existe dependencia económica, tiene un INBI como pobres, se componen por NSE D+, D y E. Presencia de NV=1,2 y la no existencia de ND y PS. La canasta básica con el gasto alimentario es muy parecidos.

Cluster 10 Presencia de inasistencia escolar en menores en su totalidad y considerados con INBI de pobre extremo.

Cluster 11 No existe la presencia de violencia en el hogar, pero existe fallecimiento familiar debido a un enfrentamiento armado o del crimen organizado y secuestro o víctima de desaparición forzada.

Cluster 12 Presencia de violencia en el hogar con la existencia de fallecimiento familiar y desaparición forzada del mismo en su totalidad.

Cluster 13 Presencia de viviendas inadecuadas sin servicios básicos y con hacinamiento, tienen un INBI de pobre y pobre extremo, está compuesta por NSE D y E. Gran presencia NV=1,2, ND=1, 2, 3, 4 y PS= 1, 2, 3.

Cluster 14 Presencia de viviendas inadecuadas sin servicios básicos y sin hacinamiento, tienen un INBI de pobre extremo, está compuesta en su totalidad por NSE E. Gran presencia NV=1 y PS=1.

Cluster 15 Presencia de viviendas inadecuadas con servicios básicos y sin hacinamiento, tienen un INBI de pobre, esta compuesta por NSE D y E. Gran presencia NV=1 y PS=1.

Cluster 16 Presencia de viviendas inadecuadas con servicios básicos y sin hacinamiento, tienen un INBI de pobre, está compuesta por NSE D+ y D. Gran presencia NV=1 y PS=1, 2, 3, 4.

Cluster 17 Presencia violencia en el hogar, sin fallecimiento y desaparición.

Cluster 18 Presencia violencia en el hogar, se realizó un tipo de acción para resolverlo, lo cual disminuye en el número de fallecimientos y desapariciones el hogar.

3.2. Segundo experimento

Para contrastar que existe tal comportamiento por parte de los datos en los nuevos grupos y que se pueda utilizar esta distribución generada para identificar la tendencia de un nuevo sujeto. Se implementaron los clasificadores Mínima

Distancia, KNN (K=1, 3, 5), una RNA entrenada con Back-Propagation y una RNA con PSO. Cada clasificador fue ejecutado 35 veces, en cada iteración se usó con un conjunto de datos de entrenamiento diferente. Se realizaron experimentos con un 20, 50 % de datos de entrenamiento de cada una de las clases.

Los Clasificadores Mínima Distancia y KNN (K= 1, 3, 5) utilizaron como métrica la distancia Minkowski de orden $\lambda = 3$. La estructura de la RNA entrenada con Back-propagation fue 25 neuronas en la capa de entrada, 49 en la capa intermedia, 18 en la capa de salida y el número de épocas igual a 10000. La RNA con PSO fue 25 neuronas en la capa de entrada, 100 partículas, 49 en la capa intermedia, 18 en la capa de salida y el número de épocas igual a 10000.

En la tabla 3 se muestran los porcentajes de clasificación mayor, medio y menor, obtenidos por cada clasificador durante las 35 ejecuciones de cada uno, en cada porcentaje de entrenamiento.

Tabla 3. Porcentajes de clasificación.

Clasificador	% de entrenamiento	% Clasificación Mayor	% Clasificación Medio	% Clasificación Menor
Distancia Minima	20.00	60.85	60.74	58.59
KNN = 1	20.00	95.82	95.38	94.90
KNN = 3	20.00	87.78	87.74	85.67
KNN = 5	20.00	61.22	60.48	60.17
RNA-BackPropagation	20.00	14.52	14.29	14.29
RNA-PSO	20.00	41.89	18.81	18.10
Distancia Minima	50.00	64.31	60.36	57.14
KNN = 1	50.00	96.80	96.71	95.38
KNN = 3	50.00	97.00	96.89	96.84
KNN =5	50.00	64.63	62.85	61.23
RNA-BackPropagation	50.00	14.29	14.29	14.29
RNA-PSO	50.00	41.91	34.91	28.58

Como se puede observar el clasificador KNN en sus 3 variantes obtuvo los mejores porcentajes de clasificación en las diferentes pruebas realizadas con 20 y 50 % de entrenamiento. Logrando un mayor porcentaje de clasificación del 97.00 % cuando K=3 en un 50 % de entrenamiento y un porcentaje menor de 60.17 % con K=5 en 20 % de entrenamiento.

La RNA-BackPropagation y RNA con PSO implementada obtuvieron los porcentajes de clasificación más bajos de todos los clasificadores. Teniendo además un costo computacional muy alto, ya que su proceso de entrenamiento y clasificación fue bastante largo.

4. Conclusiones

La elección de un vector de características para la aplicación de técnicas de agrupamiento, influye directamente en los resultados del análisis, pues los

resultados dependen en gran medida de las variables que son consideradas.

Los algoritmos de clasificación, en lo particular el KNN pueden ayudar a identificar a los nuevos beneficiarios y colocarlos con su respectiva tendencia.

Los indicadores de vivienda inadecuada, sin servicios, dependencia económica e inasistencia escolar que corresponden al INBI con pobre y pobre extremo, describen mejor los resultados y muestran una idea más clara del comportamiento en los grupos. Estos indicadores hasta el momento no son usando por el DIF, pero están descritos por el CEPAL y la FAO.

El DIF para seleccionar un posible beneficiario utiliza al MV, esta ecuación no describe el comportamiento de la población. Ya que no se logró obtener separabilidad con respecto a esta.

Se obtuvo mayor información relevante, que puede ser empleada al momento de toma decisiones en el caso de si se otorgara algún tipo de apoyo y/o beneficio.

Los grupos generados dan una visión más clara del comportamiento de la población con respecto a sus formas de vida y por ello se pueden proponer diferentes tipos de acciones que se salen del Sistema de Asistencia Alimentaria del DIF. Este tipo de acciones pueden ser, por dar un ejemplo el mejoramiento de las necesidades básicas insatisfechas en el hogar o el apoyo a la inasistencia escolar.

Se presenta una canasta básica en los sujetos por debajo de lo establecido por Consejo Nacional de Evaluación de la Política de Desarrollo Social (CONEVAL). Además que no se consumen los alimentos mínimos necesarios.

Referencias

1. Samarasinghe, S.: Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition. CRC Press., pp. 253–254 (2006)
2. Jain, A. K., Murty, M. N., and Flynn, P. J.: Data clustering: a review. ACM computing surveys (CSUR), 31(3):264–323 (1999)
3. Esbensen, K. H., Guyot, D., Westad, F., y Houmoller, L. P.: Multivariate data analysis: in practice: an introduction to multivariate data analysis and experimental design. Multivariate Data Analysis, pp. 72–74 (2002)
4. Valencia, J. L., y Diaz-Llanos, F. J.: Regresión PLS en las ciencias experimentales. Madrid: Complutense, pp. 47–55 (2003)
5. Webb, A. R.: Statistical pattern recognition. Number 501-545. John Wiley y Sons (2011)
6. Rencher, A. C.: Methods of multivariate analysis. Vol. 492, John Wiley y Sons (2003)
7. Bandyopadhyay, S., Saha, S.: Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications. Springer Science y Business Media, 8–9, 68–72 (2012)
8. Theodoridis, S., Pikrakis, A., Koutroumbas, K., and Cavouras, D.: Introduction to Pattern Recognition: A Matlab Approach: A Matlab Approach. Number 1. Academic Press (2010)
9. Bishop, C. M. et al.: Pattern recognition and machine learning. volume 1, Springer New York (2006)

10. Cuevas-Ortuño, J. and Gómez-Padilla, A. Un modelo de asignación empaque de despensas personalizadas para bancos de alimentos: un sistema sujeto a condiciones nutricionales y logísticas. *DYNA-Ingeniería e Industria*, 88(5) (2013)
11. Rodríguez, Roberto; Sossa J. Humberto.: Procesamiento y análisis digital de imágenes. 1ª Edición, Alfaomega, México, pp. 19, 155-173 (2012)
12. Theodoridis, S., y Koutroumbas, K.: *Pattern recognition*, Fourth edition, pp. 30–31 (2009)
13. Asociación Mexicana de Inteligencia de Mercado y Opinión Pública *AMAI*, Consultado Marzo 2015 <http://nse.amai.org/nseamai2/>
14. FAO: *La Seguridad Alimentaria: información para la toma de decisiones. Guía práctica*. 1–4 (2011)
15. García, C. E., y Sánchez, A. S. Clasificaciones de la OMS sobre discapacidad. *Boletín del RPD*, 50, 15–30 (2001)
16. FAO: *Ley marco derecho a la alimentación, seguridad y soberanía alimentaria. Aprobada en la XVIII Asamblea Ordinaria del Parlamento Latinoamericano*, 16–23 (2012)
17. CEPAL: *El uso de indicadores socioeconómicos en la formulación y evaluación de proyectos sociales*. 15–32 (2001)
18. de la ELCSA, C. C.: *Escala Latinoamericana y Caribeña de seguridad alimentaria (ELCSA): manual de uso y aplicación*. 9–83 (2012)
19. Consejo Nacional de Evaluación de la Política de Desarrollo Social *CONEVAL*, Consultado Marzo 2015 <http://www.coneval.gob.mx/Medicion/Paginas/Glosario.aspx> (2015)
20. Organización Mundial de la Salud *OMS*, Consultado Marzo 2015 <http://www.who.int/suggestions/faq/es/> (2015)
21. i Vives, S. S.: *Vivir y morir en Alicante: higienistas e inversiones públicas en salud (1859-1923)*. Universidad de Alicante, 22 (2008)
22. Eberhart, R. C. and Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pp. 39–43, New York, NY (1995)
23. Morales-Ruán, M., Shamah-Levy, T., Mundo-Rosas, V., Cuevas-Nasu, L., Romero-Martínez, M., Villalpando, S., Rivera-Dommarco, J. A., et al.: Food assistance programs in Mexico, coverage and targeting. *salud pública de México*, 55:S199–S205 (2013)
24. FAO: *Validación de la Escala Latinoamericana y Caribeña para la Medición de la Seguridad Alimentaria (ELCSA) en Guatemala*. Departamento de Nutrición Humana Universidad Estatal de Ohio. 1–18 (2010)
25. Jain, A. K., Duin, R. P. W., y Mao, J.: Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1), 4–37 (2000)
26. Busygin, S., Prokopyev, O., y Pardalos, P. M.: Biclustering in data mining. *Computers and Operations Research*, 35(9), pp. 2964–2987 (2008)
27. Lemaire, V., Hue, C., and Bernier, O.: Correlation analysis in classifiers. Chapter *From data mining to knowledge discovery: An overview*, pp. 1–34 (2010)
28. Eberhart, R. C., y Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micro machine and human science*, Vol. 1, pp. 39–43 (1995)
29. Solé, R. V., & Manrubia, S. C.: *Orden y caos en sistemas complejos (Vol. 2)*. Univ. Politèc. de Catalunya, 521 (2001)
30. Lemaire, V., Hue, C., and Bernier, O.: Correlation analysis in classifiers. Chapter *From data mining to knowledge discovery: An overview*. pp. 1–34 (2010)

Control difuso del parámetro β de una heurística constructiva tipo GRASP para el problema de la bisección de vértices de un grafo

Javier Alberto Rangel González, Jesús David Terán Villanueva, Héctor Joaquín Fraire Huacuja, Juan Javier González Barbosa, José Antonio Martínez Flores, Guadalupe Castilla Valdez, Apolinar Ramírez Saldivar

Instituto Tecnológico de Ciudad Madero (ITCM), Cd. Madero, Tam., México

javieralberto64@hotmail.com, david_teran01@yahoo.com.mx, automatasm2002@yahoo.com.mx, jjgonzalezbarbosa@hotmail.com, jose.mtz@itcm.edu.mx, gpe_cas@yahoo.com.mx, apolinar_r@yahoo.com

Resumen. El problema de la bisección de vértices de un grafo conexo es un problema de la clase NP-duro con importantes aplicaciones prácticas y que consiste en separar los vértices en dos conjuntos del mismo (o casi del mismo) tamaño de tal manera que se minimice la cantidad de vértices que tengan al menos una arista entre ambos conjuntos. A pesar de que actualmente se considera que los métodos heurísticos constituyen una de las herramientas más prometedoras para la solución de problemas NP-duros, en la literatura revisada no se encontraron reportes de soluciones del problema con métodos heurísticos. En éste artículo se describe una heurística constructiva de tipo GRASP, en la que el parámetro β se ajusta automáticamente usando un sistema de control difuso. Se presenta evidencia experimental de que el control difuso mejora el desempeño del algoritmo y se verifica que las diferencias observadas son estadísticamente significativas aplicando una prueba de hipótesis no paramétrica.

Palabras clave: controlador difuso, sistema difuso, control de parámetros, GRASP.

1. Introducción

El problema de la bisección de vértices de un grafo conexo es un problema de la clase NP-duro, que consiste en separar los vértices en dos conjuntos del mismo (o casi del mismo) tamaño de tal manera que se minimice la cantidad de vértices que tengan al menos una arista entre ambos conjuntos [1], [2], [3]. Este problema tiene una aplicación práctica en la distribución de información en redes de comunicaciones. Las tres principales tareas que se realizan en una red para distribuir la información son: divulgación, acumulación y *gossip* [4]. Para realizar eficientemente la divulgación, la técnica de *gossip* divide la red en dos subgrafos con aproximadamente el mismo número de nodos e identifica los nodos que conectan ambas regiones de la red. Dichos

nodos se utilizan para acumular la información en la región en que se ubican y los enlaces son utilizados para transmitir información de una región a la otra. A pesar de la relevancia práctica del problema en la literatura revisada solo se encontró un reporte en el que se proponen dos métodos exactos [5]. En dicho trabajo describen un método basado en la técnica de ramificación y acotamiento y uno basado en planos de corte y se considera que el grafo tiene un número de vértices par y que todas las aristas tienen un peso asignado. El problema consiste en separar los vértices del grafo en dos conjuntos del mismo tamaño, de tal manera que se minimice el costo total de los cruces entre dichos conjuntos. Si se considera que los costos de las aristas son uno, el problema se reduce al de la bisección de vértices. Una limitación de ese trabajo [5] es que no se aplica cuando el número de vértices es impar y que solo resuelve el problema para instancias de pequeña o en el mejor de los casos de mediana escala. Actualmente se considera que los métodos heurísticos constituyen una de las herramientas más prometedoras para la solución de instancias grandes de problemas NP-duros.

En éste artículo se describe una heurística constructiva de tipo GRASP [6] [7] en la que el parámetro β se ajusta automáticamente usando un sistema de control difuso.

2. Marco teórico

2.1. Sistemas de inferencia difusa

La lógica difusa fue propuesta por Lofthi Zadeh como una extensión de la lógica tradicional (booleana), la cual utiliza como concepto básico el grado de pertenencia (o membresía) a un conjunto difuso [8]. Los conjuntos difusos se aplican cuando no es muy clara la pertenencia de un elemento a un conjunto.

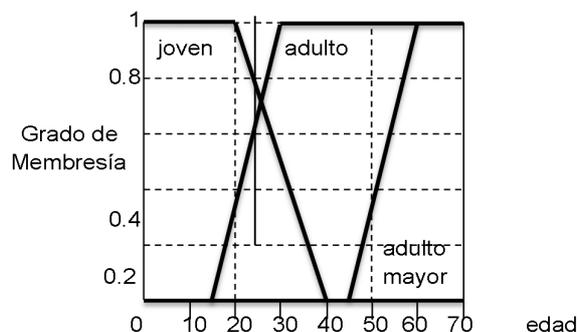


Fig. 1. Ejemplo de conjuntos difusos.

Una persona de 23 años de edad, ¿es un joven o un adulto?; una persona de 1.80 metros, ¿es alto o no?, una estrella de mar, ¿es un animal o no? Si se usa la lógica tradicional, no se podría contestar a estas preguntas de forma precisa. Sin embargo, la lógica difusa permite definir relaciones de pertenencia en una escala del cero al 1. Una persona de 23 años de edad pertenecerá al conjunto “joven” con un valor de .85

aproximadamente y al conjunto “adulto” con un valor de 0.55 aproximadamente (ver Fig. 1)

Los sistemas de inferencia difusa (FIS, por sus siglas en inglés), transforman una serie de entradas no difusas en un conjunto de salidas no difusas. Una vez realizada la transformación las salidas son utilizadas para tomar algún tipo de decisión. Para realizar el proceso estos sistemas utilizan conjuntos difusos, funciones de membresía asociadas a las entradas y las salidas, operadores lógicos difusos, reglas de inferencia difusa de tipo si-entonces y un motor de inferencia difusa. Estos sistemas han sido usados en una gran variedad de áreas tales como el control automático, la clasificación de datos, el análisis de decisiones, los sistemas expertos, la predicción de series de tiempo, la robótica y en el reconocimiento de patrones. La figura siguiente muestra la arquitectura de un FIS.

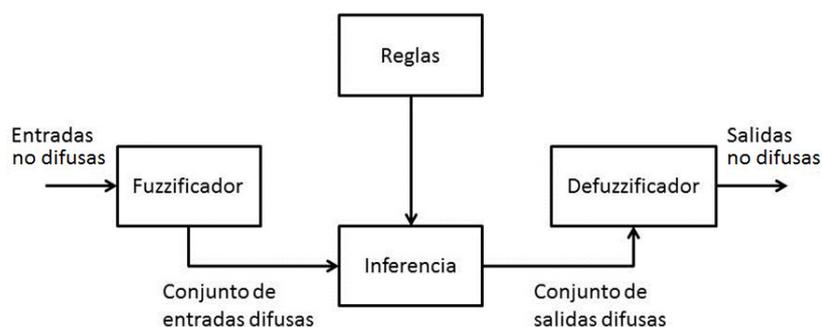


Fig. 2. Arquitectura de un sistema de inferencia difusa (FIS).

El fuzzificador utiliza funciones de membresía para transformar las entradas no difusas en valores difusos que activan el motor de inferencia. Las reglas de inferencia utilizan operadores lógicos difusos para establecer como se relacionan los valores difusos de las entradas con los valores difusos de las salidas. El motor de inferencia realiza en paralelo las transformaciones establecidas en todas las reglas de inferencia. Los modelos de tipo Mamdani y Sugeno son los que se utilizan con mayor frecuencia en la implementación del motor de inferencia [9], [10]. Finalmente el módulo de defuzzificación transforma los valores difusos de las salidas en valores no difusos.

3. Heurística constructiva tipo GRASP con control difuso

3.1. Descripción del algoritmo

La metodología GRASP fue propuesta por Feo y Resende [6], [7] y el acrónimo fue establecido por Feo et al. en [7]. El algoritmo GRASP es un algoritmo de reinicio el cual consiste principalmente de una etapa de construcción y de un algoritmo de búsqueda local. En este artículo se muestra un algoritmo constructivo tipo GRASP diseñado para el problema de Bisección de Vértices (ver Figura 3).

Al inicio del algoritmo se inicializa la cantidad de elementos seleccionados (*Elem*), todos los vértices se encuentran en el conjunto derecho y se considera que cada vértice tiene a todos sus adyacentes (*Grado_v*) en el lado derecho (*AdyToR_v*). El primer elemento seleccionado será aquél con menor grado, aquél que tenga menos enlaces a elementos de la derecha, este elemento será removido del conjunto derecho y asignado al izquierdo. Además, se realizará una actualización de todos los adyacentes del elemento seleccionado, esto se hace con el fin de identificar aquéllos vértices que tienen menos conexiones hacia el conjunto de la derecha (*AdyToR*). Después se creará la lista de candidatos (*LC*) con todos los elementos que todavía se encuentran en el conjunto de la derecha (*R*). Y la lista de candidatos restringida (*LCR*) contendrá los mismos elementos que *LC*, menos aquellos elementos que tengan más adyacentes a la derecha que los definidos por la variable Límite. Siendo el Límite un valor entre el mínimo y máximo número de adyacentes a la derecha de todos los vértices en *LC*, el valor del límite depende de la variable β que es un valor entre cero y uno. Si la variable β tiende a cero entonces el límite será muy restrictivo, mientras que si β tiende a uno entonces el límite será muy laxo. Luego se selecciona un vértice (*u*) de *LCR* de manera aleatoria, se elimina del conjunto de la derecha (*R*) y se incluye en el conjunto de la izquierda (*L*). Posteriormente se actualizan los adyacentes hacia la derecha *AdyToR* disminuyendo en uno a todos los vértices que son adyacentes a *u*. Se continúa con este proceso hasta que se hayan movido, al conjunto de la izquierda, la mitad de vértices del grafo.

```

Elem = 0
AdyToRv = Gradov      ∀v ∈ V
R ← {v}                ∀v ∈ V
L = ∅
u = ArgMinv(Gradov)
L ← {u}
R = R \ {u}
AdyToRv--            ∀v ∈ V | ∃(u, v) ∈ E
Elem++
do{
  LC = R
  Limite = Min(AdyToRv) + β (Max(AdyToRv) -
Min(AdyToRv))   ∀v ∈ LC
  LCR = LC \ v ∈ LC | AdyToRv > Limite
  u = SelectAleatoria(LCR)
  L ← {u}
  R = R \ {u}
  AdyToRv--            ∀v ∈ V | ∃(u, v) ∈ E
  Elem++
}while(Elem < [|V|/2])

```

Fig. 3. Algoritmo constructivo tipo GRASP.

3.2. Controlador difuso del parámetro β

Al algoritmo anterior se le incorporó un controlador difuso para ajustar automáticamente el parámetro β . El controlador difuso recibe como variable de entrada no difusa la β y el $\%Err$ de la solución actual con respecto a la mejor observada en el proceso de búsqueda y como variable de salida no difusa $\Delta\beta$, el cual especifica un incremento en la β .

En cada iteración el algoritmo genera la solución actual, se actualiza la mejor solución encontrada y se determina el porcentaje de error ($\%Err$). Con la β y $\%Err$ se invoca el controlador difuso, para determinar el valor de $\Delta\beta$, actualizar el valor de β y realizar la siguiente iteración.

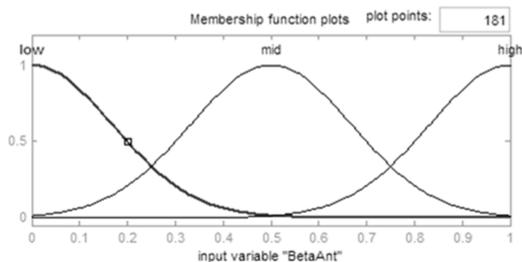


Fig. 4. Variable de entrada “ β ”

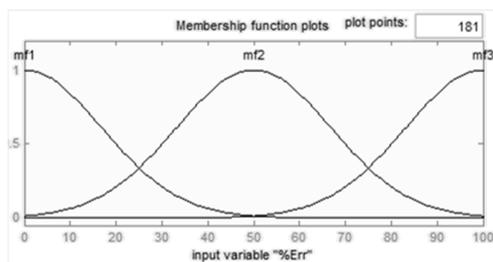


Fig. 5. Variable de entrada “ $\%Err$ ”

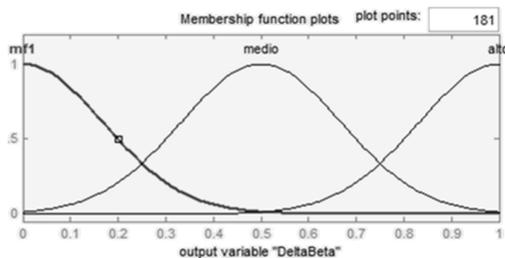


Fig. 6. Variable de salida “ $\Delta\beta$ ”

Las entradas del controlador son β y $\%Err$ y la salida es $\Delta\beta$. El controlador usa 3 funciones de membresía gaussianas (bajo, medio, alto) tanto para el parámetro de

entrada como el de salida. Estas funciones de membresía tienen la característica de ser simétricas y con traslape. Las siguientes figuras (Figura 4, Figura 5 y Figura 6) muestran las funciones de membresía utilizadas.

Las reglas que utiliza el controlador son las siguientes:

1. Sí, β es bajo y $\%Err$ es bajo entonces $\Delta\beta$ será alto;
2. Sí, β es bajo y $\%Err$ es medio entonces $\Delta\beta$ será medio;
3. Sí, β es bajo y $\%Err$ es alto entonces $\Delta\beta$ será alto;
4. Sí, β es medio y $\%Err$ es bajo entonces $\Delta\beta$ será bajo;
5. Sí, β es medio y $\%Err$ es medio entonces $\Delta\beta$ será bajo;
6. Sí, β es medio y $\%Err$ es alto entonces $\Delta\beta$ será alto;
7. Sí, β es alto y $\%Err$ es bajo entonces $\Delta\beta$ será medio;
8. Sí, β es alto y $\%Err$ es medio entonces $\Delta\beta$ será alto;
9. Sí, β es alto y $\%Err$ es alto entonces $\Delta\beta$ será medio;

El motor de inferencia que utiliza el controlador es de tipo Mamdani.

4. Resultados experimentales

Toda la experimentación se realizó en una computadora portátil con las siguientes características: sistema operativo Windows 7 de 64 bits con service pack 1, procesador Intel core i7 a 3.1 GHz y memoria RAM de 8 GB.

El algoritmo constructivo tipo GRASP fue implementado en C++, y se implementaron dos versiones del controlador difuso. Una utiliza una librería desarrollada por los autores en Java y la otra utiliza la librería disponible en Matlab.

En los experimentos realizados se comparan los siguientes algoritmos: uno en el que el parámetro β tiene un valor fijo de .5, otro en el que el parámetro β se ajusta con el controlador implementado en Java, y otro en el que el parámetro β se ajusta con el controlador implementado en Matlab. Para comparar el desempeño de estos algoritmos se resolvieron con cada uno de ellos 87 instancias Harwell-Boeing [11]. Estas constituyen el conjunto de instancias más difíciles con las que se evalúa el desempeño de algoritmos heurísticos para problemas de ordenamiento lineal.

Tabla 1. Resultado de las instancias Harwell-boeing.

Instancias	Sin controlador difuso	Java	Matlab
494_bus.mtx.rnd	194	161	154
662_bus.mtx.rnd	289	226	230
685_bus.mtx.rnd	328	257	259
arc130.mtx.rnd	65	33	29
ash292.mtx.rnd	145	129	127
ash85.mtx.rnd	41	35	33
bcspwr01.mtx.rnd	16	10	7

Instancias	Sin controlador difuso	Java	Matlab
bcsplr02.mtx.rnd	19	13	10
bcsplr03.mtx.rnd	55	36	37
bcsplr04.mtx.rnd	119	102	99
bcsplr05.mtx.rnd	192	149	148
bcsstk01.mtx.rnd	24	23	20
bcsstk02.mtx.rnd	33	33	33
bcsstk04.mtx.rnd	66	66	66
bcsstk05.mtx.rnd	75	73	74
bcsstk06.mtx.rnd	197	199	200
bcsstk20.mtx.rnd	190	187	195
bcsstk22.mtx.rnd	49	43	37
bcsstm07.mtx.rnd	199	201	204
can__144.mtx.rnd	72	63	59
can__161.mtx.rnd	80	71	75
can__292.mtx.rnd	144	126	124
can__445.mtx.rnd	220	206	209
curtis54.mtx.rnd	26	20	17
dwt__209.mtx.rnd	103	92	91
dwt__221.mtx.rnd	110	98	98
dwt__234.mtx.rnd	56	34	31
dwt__245.mtx.rnd	113	98	97
dwt__310.mtx.rnd	154	143	138
dwt__361.mtx.rnd	179	166	162
dwt__419.mtx.rnd	209	197	196
dwt__503.mtx.rnd	248	240	241
dwt__592.mtx.rnd	295	276	276
fs_183_1.mtx.rnd	83	67	66
fs_541_1.mtx.rnd	262	263	257
fs_680_1.mtx.rnd	308	240	234
gent113.mtx.rnd	50	43	47
gre_216a.mtx.rnd	104	98	99
gre__115.mtx.rnd	54	43	44
gre__185.mtx.rnd	91	86	82
gre__343.mtx.rnd	164	159	156
gre__512.mtx.rnd	247	240	241
hor__131.mtx.rnd	215	204	206
ibm32.mtx.rnd	15	13	13

Instancias	Sin controlador difuso	Java	Matlab
impcol_a.mtx.rnd	100	86	88
impcol_b.mtx.rnd	25	26	24
impcol_c.mtx.rnd	67	55	52
impcol_d.mtx.rnd	209	188	185
impcol_e.mtx.rnd	106	104	100
lns__131.mtx.rnd	54	42	40
lns__511.mtx.rnd	228	206	201
lund_a.mtx.rnd	70	70	71
lund_b.mtx.rnd	70	70	72
mbeacxc.mtx.rnd	243	243	242
mcca.mtx.rnd	81	75	75
nnc261.mtx.rnd	119	106	111
nnc666.mtx.rnd	313	282	295
nos1.mtx.rnd	76	61	58
nos2.mtx.rnd	314	268	275
nos4.mtx.rnd	45	38	41
nos5.mtx.rnd	233	230	229
nos6.mtx.rnd	330	279	286
plat362.mtx.rnd	181	178	174
plskz362.mtx.rnd	179	149	149
pores_1.mtx.rnd	13	13	13
pores_3.mtx.rnd	217	211	212
saylr1.mtx.rnd	116	97	98
saylr3.mtx.rnd	333	275	277
sherman4.mtx.rnd	266	236	244
shl__200.mtx.rnd	288	292	285
shl__400.mtx.rnd	286	291	285
shl____0.mtx.rnd	288	287	280
steam1.mtx.rnd	120	117	117
steam2.mtx.rnd	300	300	300
steam3.mtx.rnd	40	36	36
str__200.mtx.rnd	167	172	173
str__600.mtx.rnd	170	171	175
str____0.mtx.rnd	166	164	167
west0132.mtx.rnd	65	52	56
west0156.mtx.rnd	76	67	65
west0167.mtx.rnd	81	71	73
west0381.mtx.rnd	189	184	183

Instancias	Sin controlador difuso	Java	Matlab
west0479.mtx.rnd	227	222	221
west0497.mtx.rnd	238	214	219
west0655.mtx.rnd	313	311	307
will199.mtx.rnd	96	91	91
will57.mtx.rnd	24	17	18
<i>Promedios</i>	149.6551724	136.88505	136.597701

En estos resultados se observa que en casi todos los casos se reduce o se empata el valor objetivo al emplear un controlador difuso, ya sea en Java o Matlab. También podemos ver que se reduce el valor promedio de la función objetivo al emplear cualquier variante de los controladores difusos.

Los resultados de la prueba no paramétrica de Wilcoxon muestran que las mejoras observadas al utilizar los controladores difusos son estadísticamente significativas (con una certidumbre superior al 99%). Por otra parte, la prueba de Wilcoxon muestra resultados de equivalencia estadística entre los resultados obtenidos con la aplicación en Java y los resultados de Matlab. Estos resultados muestran que la implementación en Java posee la misma calidad que Matlab, y por lo tanto es una alternativa viable para realizar pruebas de sistemas difusos.

5. Conclusiones

En éste artículo se describe una heurística constructiva de tipo GRASP, en la que el parámetro β se ajusta automáticamente usando un sistema de control difuso. La evidencia experimental muestra que el control difuso mejora el desempeño del algoritmo y se verifica que las diferencias observadas son estadísticamente significativas. Se considera que este enfoque puede ser aplicado para controlar el parámetro β de heurísticas tipo GRASP aplicadas a otros problemas.

Referencias

1. Díaz, J., Petit, J., & Serna, M.: A Survey on Graph Lay-out Problems. *CM Computing Surveys (CSUR)*, 34(3), pp. 313–356 (2000)
2. F. D. Brandes Ulrik: Vertex Bisection is Hard, too. *Journal of Graph Algorithms and Applications* (2009)
3. Fraire, Héctor, et al.: Exact Methods for the Vertex Bisection Problem. *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*. Springer International Publishing. pp. 567–577 (2014)
4. Hans-Joachim, B.: Two open problems in communication in edge-disjoint paths modes. *Acta Mathematica et Informatica Universitatis Ostraviensis*, Vol. 7, No. 1 (1999)
5. Karisch, S. E., Rendl, F., Clausen, J.: Solving graph bisection problems with semidefinite programming. *INFORMS Journal on Computing*, 12(3), pp. 177–191 (2000)

6. Feo, T. A., Resende, M. G.: A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2), pp. 67–71 (1989)
7. Feo, T. A., Resende, M. G.: Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2), pp. 109–133 (1995)
8. Zadeh, L.: Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 1, pp. 28–44 (1973)
9. Mamdani, E. H., & Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1), pp. 1–13 (1975)
10. Sugeno, M.: Fuzzy measures and fuzzy integrals: a survey. *Fuzzy Automata and Decision Processes*, pp. 89–102 (1977)
11. Rojas, R.: *Neural Networks: A systematic introduction*. Springer, pp. 289–291 (1996)

Generación de señales de electroestimulación percutánea para terapias pasivas en rehabilitación neuromuscular de tobillo empleando lógica difusa

Miriam C. Reyes-Fernández, Ruben Posada-Gomez, Albino Martinez-Sibaja,
Alberto A. Aguilar-Lasserre, Juan C. Trujillo-Caballero

División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Orizaba
Orizaba, Veracruz, México

mcrefer@gmail.com, rposada@itorizaba.edu.mx,
albino3_mx@yahoo.com, albertoaal@hotmail.com,
juan_carlostrujillo@hotmail.com

Resumen. Este artículo describe el desarrollo de un sistema de electroestimulación percutánea para terapias pasivas, que pueda mejorar los procesos de rehabilitación aplicados a los pacientes con esguince en el tobillo, mediante el empleo de señales de electroestimulación para mejorar el impacto de los ejercicios y favorecer la retroalimentación entre el profesional y el paciente, logrando de esta manera una recuperación más rápida y eficaz. Se plantea el empleo de la Lógica Difusa como herramienta para determinar el tipo de señales más convenientes a usar en el tratamiento del paciente tomando la experiencia del especialista dependiendo de las características del paciente así como del tipo de lesión. Así mismo se sientan las bases para el desarrollo de los protocolos de experimentación correspondientes para algunos casos de estudio.

Palabras clave: Rehabilitación pasiva, electroestimulación percutánea, rehabilitación de tobillo.

1. Introducción

Las lesiones de tobillo son comunes a lo largo de la vida de un ser humano, sin importar condiciones de vida, si hacen o no deporte, edad, peso, etc. Ocurren en cualquier momento, y suelen ser dolorosas y de recuperación a mediano o largo plazo. Según la Secretaría del Trabajo y Previsión Social [1], las lesiones del tobillo y pie son el segundo tipo de lesiones más frecuentes después de las lesiones de muñeca y mano (ver Fig. 1).

Con el fin de evitar que las lesiones que una persona sufra lo lleven a una discapacidad motora permanente, es necesario que esta lleve un proceso de rehabilitación. El cual, dependiendo del tipo de lesión consiste en la una serie de ejercicios y tratamientos diseñados por el médico traumatólogo en conjunto con un equipo de terapeutas en rehabilitación motriz. Los tratamientos sugeridos pueden clasificarse de manera global en dos tipos [2]:

- **Rehabilitación activa:** Consiste en una serie de ejercicios que el paciente debe hacer permitiendo que se realice el movimiento de las articulaciones afectadas, el terapeuta orienta al paciente para la realización de la rutina y generalmente se realiza precozmente tras los primeros días del episodio, generalmente en centros especializados.

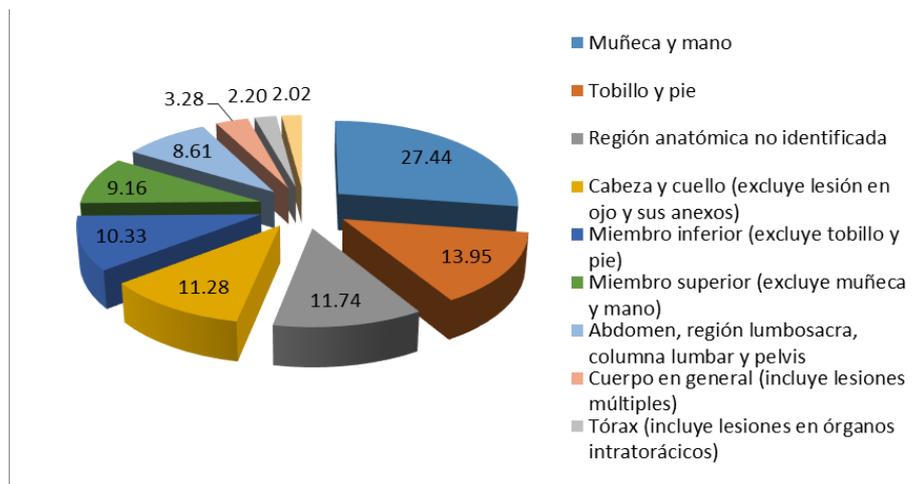


Fig. 1. Accidentes de Trabajo Según Región Anatómica y Sexo, 2010 – 2012 Nacional (Datos obtenidos de [1]).

- **Rehabilitación pasiva:** Consiste en la movilización por parte del terapeuta de las articulaciones afectadas tres o cuatro veces al día, para prevenir contracturas musculares y lesiones articulares.

En ambos casos se cuentan con diferentes herramientas que ayudan a la realización de este tipo de terapias. En el caso de la rehabilitación activa, la mecanoterapia es la herramienta más frecuentemente utilizada [3]; Por otro lado, en el caso de la rehabilitación pasiva se reporta que los movimientos asistidos produzcan una cierta relajación de las estructuras fibrosas y de esta manera se mantienen activos las estructuras afectadas reduciendo las contracturas y retracciones musculares, en estas circunstancias recientemente se han efectuado trabajos que sugieren que la electroestimulación es una herramienta recomendable para recuperar la movilidad [4].

De la misma manera tanto la terapia de rehabilitación activa como la pasiva presentan diferentes ventajas para la recuperación de la movilidad, sin embargo estas técnicas de rehabilitación se aplican de manera alternada. En los casos en los que el paciente tiene muy poca movilidad o no es capaz de efectuar los movimientos por sí mismo se aplica la rehabilitación pasiva y el uso de la electroestimulación juega el papel de prevención de lesiones articulares por inmovilidad; cuando el paciente tiene movilidad suficiente para efectuar los movimientos por sí mismo, continua su tratamiento en una sala de mecanoterapia, donde los aparatos mecánicos tienen la función de limitar el movimiento que se efectúa durante las terapias.

2. Lesiones del tobillo

El tobillo es la parte de la anatomía humana donde se unen el pie y la pierna. La pieza superior está conformada por la tibia y el peroné, huesos de la pierna, y por el astrágalo, que forma parte del pie.

El tobillo es el responsable de la marcha y de los movimientos que se hacen al andar así como de permitir que el ser humano pueda mantener una postura erguida.

Al tener una función tan importante para el movimiento es común que esta articulación sufra lesiones que van de leves a severas, ya que continuamente está expuesta a diversos accidentes que derivan en fracturas o esguinces [5]. Estas lesiones conllevan dolor, inmovilización y un tiempo de recuperación que varía de acuerdo al tipo, a las condiciones del paciente y a las condiciones de rehabilitación con que se cuenten, en este sentido existen diversos estudios enfocados a este tipo de lesiones principalmente en el ámbito de la medicina del deporte [6].

Una persona puede sufrir un esguince de tobillo a cualquier edad y casi cualquier circunstancia de marcha, sin embargo, hay grupos con más riesgo, como los deportistas o las personas de la tercera edad; los primeros, porque someten sus cuerpos a pruebas físicas extremas, los segundos, porque el sistema óseo se encuentra deteriorado y muchas veces también está resentida la articulación por el paso del tiempo y por accidentes previos.

2.1. Tipos de lesiones

El tobillo puede resistir impactos, pero cuando exceden el límite de resistencia de los ligamentos o de los huesos, se presentan fracturas o esguinces que deben ser atendidas inmediata y adecuadamente, de lo contrario, podrían presentarse secuelas que podrían incluso llevar a una discapacidad permanente.

En una lesión del tobillo intervienen tanto la dirección como la magnitud de las fuerzas y la posición del pie y el tobillo durante el traumatismo generalmente en flexión plantar por lo que la inflamación y el dolor suelen estar presentes en el Fascículo Peroneo-astragalino Anterior (FPAA), en el Fascículo Peroneo-calcáneo (FPC) y el Fascículo Peroneo-astragalino Posterior (FPAP) [7].

Tabla 1. Sistema de clasificación para la descripción de los grados de esguince del ligamento lateral externo del tobillo.

Características clínicas	Grado 1	Grado 2	Grado 3
Localización del dolor	FPAA	FPAA,FPC	FPAP,FPC,FPAA
Edema, Equimosis	Local y escasamente notable	Local y moderadamente notable	Difuso (varios segmentos del complejo articular) muy notable
Capacidad de carga	Completa o parcial	Difícil sin muletas	Incapacidad funcional total
Daño ligamentoso	Elongación	Ruptura parcial	Ruptura total
Inestabilidad	No presenta	Moderada (escasa) o no presenta	Inestabilidad total

El esguince de tobillo es una de las lesiones más frecuentes, consiste en un estiramiento que hace que el ligamento supere su límite de elasticidad. Un esguince se produce por apoyar mal el pie, caídas o flexiones muy fuertes, movimientos bruscos al hacer deportes, y una enorme variedad de movimientos que pueden provocarla. Esto produce que se estiren los ligamentos de manera extraordinaria provocando inflamación, hematomas y la dificultad para andar o ponerse de pie. La Tabla 1, muestra la clasificación según el grado de esguince en el tobillo [8].

La rehabilitación de un esguince suele ser más larga de lo que la mayoría de los pacientes espera. El período de recuperación tras el tiempo de reposo suele ser en promedio de 15 días para los esguinces de grado I; 44 días para el grado II; y de 62 días para los esguinces de grado III [7].

2.2. Rehabilitación del tobillo

Una vez que el tobillo se ha lesionado, lo primero es inmovilizar la articulación para evitar afectaciones mayores, acto seguido es necesaria la evaluación médica para descartar (o confirmar) fracturas.

De confirmarse el diagnóstico de esguince en el tobillo, se procede a evaluar el grado del mismo, de acuerdo a la siguiente clasificación:

- Grado I o leve: hay estiramiento de las fibras y lesiones microscópicas. No existe inestabilidad de la articulación y, en general permite caminar con molestias.
- Grado II, con rotura parcial de algunas fibras. Generalmente el dolor y la hinchazón aparecen rápido, y se acompaña de dolor moderado o severo; existe cierta inestabilidad y es imposible mantener el equilibrio. Es frecuente que aparezca una equimosis, un hematoma en la parte lateral del pie a los pocos días, como consecuencia del sangrado al romperse las fibras.
- Grado III, con rotura completa del ligamento, y en ocasiones con arrancamiento óseo. Existe dolor severo, inflamación importante, inestabilidad e incapacidad para el apoyo. Requiere atención por el traumatólogo y en algunos casos incluso cirugía.

Una vez determinado el tipo de esguince, se procede a inmovilizar mediante férula al tobillo, y la prescripción suele ser reposo, medicamentos contra la inflamación y el dolor y regreso a consulta médica.

Después de un tiempo que varía de acuerdo al grado de la lesión, el paciente puede comenzar a hacer movimientos de estiramiento en casa, para obtener de nuevo fuerza y tonicidad muscular. De ser necesario, también estará recibiendo la ayuda del experto para usar máquinas de movimiento pasivo.

Un tobillo con esguince puede requerir entre 2 y 6 semanas para recobrase, pero los esguinces más graves pueden tardar en sanar hasta 12 semanas.

3. Desarrollo del electroestimulador

La electroestimulación es una técnica que se emplea de manera paralela al tratamiento de rehabilitación motriz principalmente para la reducción de la espasticidad. Lisinski et al [9] presentan un estudio de pacientes que sufrieron un

accidente cerebrovascular y que han sido tratados mediante la electroestimulación y muestran la efectividad del tratamiento para la tonicidad como para la fuerza muscular las cuales son introducidas en la muñeca y tobillo del paciente. Para evaluar la actividad muscular se emplearon las mediciones de electromiograma encontrándose un aumento significativo en la mayoría de los pacientes.

Amiridis et al [10] presentan un estudio realizado en adultos mayores empleando electroestimulación para determinar el efecto de la flexión dorsal del tobillo. Se monitoreo el comportamiento de los pacientes después de efectuar las sesiones de electroestimulación para efectuar tareas de balance como estar parados con una sola pierna, y posturas normales cuantificando las variaciones que se obtenían en el rango de oscilación de las personas. Los autores concluyen en base a estos datos que las sesiones de electroestimulación son benéficas para pacientes no muestran una discapacidad unilateral como en los pacientes que han sufrido algún accidente cerebrovascular.

La utilización de electroestimulación con fines terapéuticos en el área de la rehabilitación continúa siendo un tema de interés para el desarrollo de prototipos como lo muestra la patente de Skahan et al [11] en la cual se presenta un sistema de electroestimulación alojado en un guante. El objetivo es que el guante desarrollado sirva al mismo tiempo como un sistema de electroestimulación y pueda transmitir las señales de acoplamiento con el usuario.

3.1. Parámetros de la electroestimulación

Para llevar a cabo la estimulación eléctrica neuromuscular existen diferentes parámetros de estimulación tales como la intensidad, forma de onda, anchura de pulso, frecuencia, etc. que son variables que deben ser tomadas en cuenta para la reproducción de los resultados esperados, ya que estos parámetros influirán considerablemente en los efectos musculares. Así por ejemplo Snyder-Mackler et al [12] reportan un aumento de fuerza al emplear señales de 75 Hz, diferente a la reportada por Lieber et al. [13] y Paternostro-Sluga et al. [14] Quienes emplean pulsos de frecuencias de entre 30-50Hz.

Para este trabajo se han considerado efectuar variaciones en la frecuencia de los pulsos determinada directamente por el número de pulsos y la duración de los mismos. Posteriormente se pretende efectuar diferentes pruebas modificando la amplitud de los pulsos.

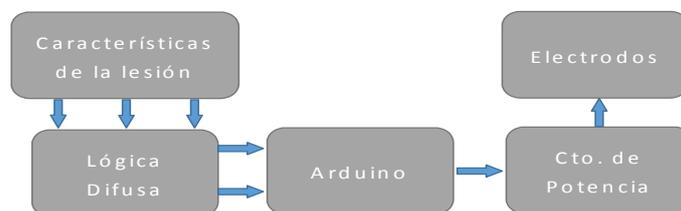


Fig. 2. Diagrama a bloques del sistema de electroestimulación desarrollado.

El electroestimulador desarrollado (Figura 2), genera los pulsos de voltajes elevados y bajas corrientes hacia los electrodos. Toma como entradas el número de pulsos y la

duración de los mismos. Estos parámetros son proporcionados por el sistema difuso que genera estos datos a partir de la información del paciente respecto al dolor que refieran (*Dolor*), su capacidad de carga (*Cap.Carga*) y la estabilidad al caminar (*Estabilidad*).

De esta manera el sistema difuso que se desarrolló (Figura 3) toma como entradas las variables que representan el dolor, la capacidad de carga y la estabilidad durante la marcha, y como salidas el número de pulsos (*N.Pulsos*) y la duración de los pulsos (*T.A.*).

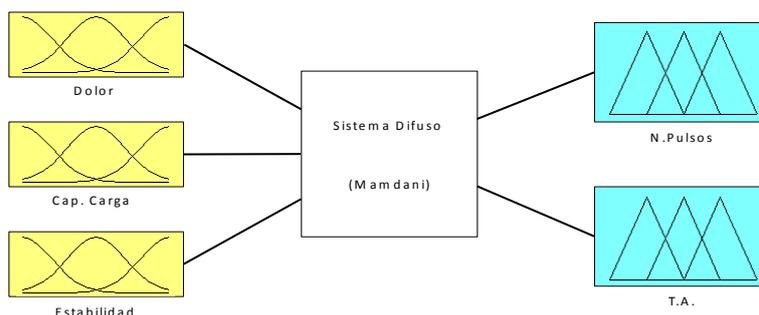


Fig. 3. Sistema difuso desarrollado.

3.2. Definición de las variables lingüísticas

El modelo de lógica difusa para la toma de decisiones tiene como propósito tomar una decisión basada en un conjunto de reglas (motor de inferencia) dadas por la experiencia del terapeuta de acuerdo a los valores de cada variable de entrada. En este caso las variables identificadas como entrada al modelo de lógica difusa son:

- **Dolor.** Al inicio de la sesión, el paciente le refiere al terapeuta el dolor que percibe en el miembro a tratar en un rango de 0 a 10, el cual puede ser identificado como “leve”, “moderado” o “agudo”. Esta entrada es definida como gaussiana con una apertura de 1.5 y centrada en 0, 5 y 10 respectivamente.
- **Cap.Carga.** El terapeuta evalúa la capacidad del paciente para sostenerse de pie mientras sostiene su propio peso en un rango de 0 a 10, esta variable se define de tipo gaussiana con una apertura de 1.5 y se identifica como “completa” (0), si el paciente puede mantenerse de pie sin dificultad, “dificultosa” (5) si lo hace refiriendo molestias o “imposibilidad” (10) si no es posible que se sostenga de pie.
- **Estabilidad.** Para determinar la estabilidad, el terapeuta solicita al paciente que camine mientras el terapeuta observa la forma en que efectúa su marcha. El rango de esta variable de tipo gaussiana se encuentra entre 0 y 10y se identifica como “S/Problemas” (0) si el paciente puede desplazarse sin ayuda, “C/Dificultad” (5) si requiere de algún tipo de ayuda para desplazarse e “Imposibilidad” (10) si el paciente no puede desplazarse.

De la misma manera, los parámetros obtenidos por las variables de inferencia generan dos tipos de salidas difusas:

- N.Pulsos. El tratamiento que el terapeuta efectuará en el paciente con el electroestimulador, envía un número de pulsos que pueden ir de 0 a 50 y son definidos como “pocos” (de 0 a 15), “medios” (15 a 35) o “muchos” (35 a 50). Esta salida es de tipo triangular con un máximo en 7.5, 25 y 42.5. El valor obtenido es redondeado al entero más cercano para ser aplicado por el sistema electroestimulador.

- T.A. Es el tiempo que el pulso estará en Alto, y este puede se encuentra en un rango de entre 0 y 250 μ s. se define como “MuyCorta” (0 a 50), “Corta” (50 a 100), “Media” (100 a 150), “Larga” (150 a 200) y “MuyLarga” (200 a 250). De la misma manera el valor de salida debe ser redondeado al entero más próximo para poder ser empleado en el electroestimulador.

El mecanismo de inferencia toma los grados de pertenencia de cada entrada y genera una salida difusa en base al conjunto de reglas difusas (Tabla 2) resumidas en forma lingüística y que contienen el conocimiento del experto. Se toman en cuenta solo aquellas condiciones que el experto considera factibles y válidas. De esta manera por ejemplo no se evalúan aquellas condiciones en los que el paciente no tiene estabilidad y pueda tener una capacidad de carga completa, dado que es una condición que no se encuentra en la realidad.

Tabla 2. Reglas de inferencia empleadas en el sistema difuso.

#	Regla	Salida
1	Dolor=Leve & Cap.Carga=Comp. & Estabilidad=S/Prob.	N.Pulsos => Pocos, T:A => MuyCorta
2	Dolor=Leve & Cap.Carga=Comp. & Estabilidad=C/Dif.	N.Pulsos => Pocos, T:A => Corta
3	Dolor=Leve & Cap.Carga=Dif. & Estabilidad=C/Dif.	N.Pulsos => Pocos, T:A => Media
4	Dolor=Leve & Cap.Carga=Dif. & Estabilidad=Imp.	N.Pulsos => Pocos, T:A => Larga
5	Dolor=Leve & Cap.Carga=Imp. & Estabilidad=C/Dif.	N.Pulsos => Pocos, T:A => Larga
6	Dolor=Leve & Cap.Carga=Imp. & Estabilidad=Imp.	N.Pulsos => Pocos, T:A => MuyLarga
7	Dolor=Leve & Cap.Carga=Comp. & Estabilidad=S/Prob.	N.Pulsos => Medios, T:A => MuyCorta
8	Dolor=Leve & Cap.Carga=Comp. & Estabilidad=C/Dif.	N.Pulsos => Medios, T:A => Corta
9	Dolor=Leve & Cap.Carga=Dif. & Estabilidad=C/Dif.	N.Pulsos => Medios, T:A => Media
10	Dolor=Leve & Cap.Carga=Dif. & Estabilidad=Imp.	N.Pulsos => Medios, T:A => Larga
11	Dolor=Leve & Cap.Carga=Imp. & Estabilidad=C/Dif.	N.Pulsos => Medios, T:A => Larga
12	Dolor=Leve & Cap.Carga=Imp. & Estabilidad=Imp.	N.Pulsos => Medios, T:A => MuyLarga
13	Dolor=Leve & Cap.Carga=Comp. & Estabilidad=S/Prob.	N.Pulsos => Muchos, T:A => MuyCorta
14	Dolor=Leve & Cap.Carga=Comp. & Estabilidad=C/Dif.	N.Pulsos => Muchos, T:A => Corta
15	Dolor=Leve & Cap.Carga=Dif. & Estabilidad=C/Dif.	N.Pulsos => Muchos, T:A => Media
16	Dolor=Leve & Cap.Carga=Dif. & Estabilidad=Imp.	N.Pulsos => Muchos, T:A => Larga
17	Dolor=Leve & Cap.Carga=Imp. & Estabilidad=C/Dif.	N.Pulsos => Muchos, T:A => Larga
18	Dolor=Leve & Cap.Carga=Imp. & Estabilidad=Imp.	N.Pulsos => Muchos, T:A => MuyLarga

Los resultados de la evaluación de las reglas se interpretan para obtener una salida con un valor preciso. Para ello se emplea el modelo del centro de gravedad. Como puede verse en la figura 4. Si se toma un caso en el que el paciente presente Dolor moderado con un valor de 5; Tenga una capacidad de carga dificultosa con un valor de

5 y presente dificultad para su estabilidad también con un valor de 5. El sistema genera una salida correspondiente a 25 pulsos con una duración de 125 μ s.

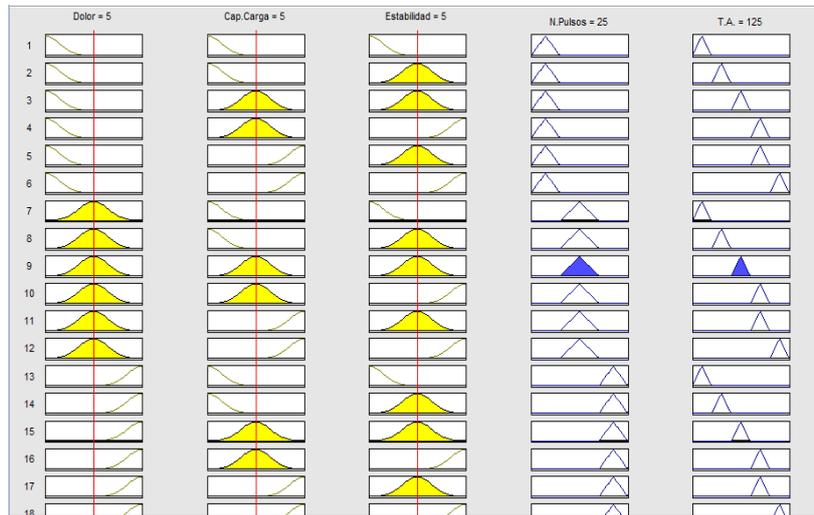


Fig. 4. Sistema difuso desarrollado.

4. Resultados obtenidos

Con el sistema desarrollado se generaron las señales de electroestimulación para diferentes casos. Los pulsos generados se muestran en la figura 5 y el código implementado en el Arduino puede verse en el Algoritmo 1.

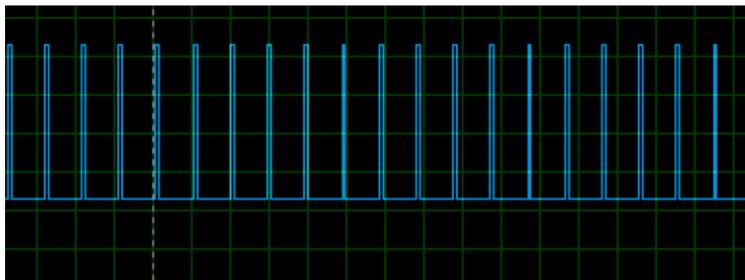


Fig. 5. Pulsos generados por el electroestimulador.

Algoritmo 1. Frangmento del programa para el Arduino empleado en el electroestimulador desarrollado.

```
int psal=13; // Puerto de salida empleado
int ta1=125; // tiempo en alto del pulso en us
int tb1=125; //tiempo en bajo del pulso us
```

```
int ctal=50; // numero de pulsos
void setup() {
  pinMode(psal,OUTPUT);
}
void loop() {
  for (int i=1; i<=ctal; i++) {
    digitalWrite(psal,HIGH);
    delayMicroseconds(tal);
    digitalWrite(psal,LOW);
    delay(tbl);
  }
}
```

5. Conclusiones y trabajos futuros

La electroestimulación es una herramienta que puede resultar de gran utilidad en la rehabilitación motriz de los pacientes, es importante sin embargo que se efectúen mayores estudios respecto a los resultados que esta técnica producen en los pacientes, ya que se ha visto mejoría sobre todo en lo concerniente a la reducción del dolor referida por los pacientes. Es necesario que se continúe con trabajos que permitan validar el empleo de la electroestimulación de una manera objetiva. El empleo de la lógica difusa resulta muy adecuado ya que permite decidir las características del tratamiento que se dará al paciente en cuanto a la duración y amplitud de las señales, tomando como base el conocimiento del experto y la sintomatología del paciente.

Como trabajos futuros, se implementará esta metodología también en la amplitud de la señal y se llevará a cabo un estudio que permita efectuar un seguimiento en los pacientes que empleen la electroestimulación con determinadas lesiones del tobillo.

Referencias

1. STPS, Información sobre Accidentes y Enfermedades de Trabajo Nacional 2003-2012. México (2012)
2. D. K. Spierer, R. Goldsmith, D. A. Baran, K. Hryniewicz, and S. D. Katz: Effects of active vs. passive recovery on work performed during serial supramaximal exercise tests. *Int. J. Sport. Med.*, vol. 25, no. 2, pp. 109–114 (2004)
3. Araújo, M. E. A. D., Silva, E. B. D., Vieira, P. C., Cader, S. A., Mello, D. B. D., Dantas, E. H. M.: Reduction of the chronic pain associated to the scoliosis non structural in university students submitted to the Pilates method. *Motriz: Revista de Educação Física*, 16(4), pp. 958–966 (2010)
4. Finberg, M., Braham, R., Goodman, C., Gregory, P., Peeling, P.: Effects of electrostimulation therapy on recovery from acute team-sport activity. *Int J Sports Physiol Perform*, 8(3), pp. 293–299 (2013)
5. Ciro, J. A. O., Rodríguez, M. P. C., E. A. V., Giraldo, S. P., Ching, I. C. G.: Lesiones deportivas. *Iatreia Rev. médica Univ. Antioquia*, no. 2, pp. 167–177 (2007)

6. García-Soidán, J., Arufe-Giraldes, V.: Analysis of the most frequent injuries in tests of speed, half and long distances. *Int. J. Med. Sci. Phys. Act. Sport*, vol. 3, no. 12, pp. 260–270 (2003)
7. Arbizu, R. L. T., Raventós, K. E., & Urrialde, J. M.: Actualización en el tratamiento fisioterápico de las lesiones ligamentosas del complejo articular del tobillo. *Fisioterapia*, 28(2), pp. 75–86 (2006)
8. CENETEC: Guía de Práctica Clínica del Diagnóstico y Manejo del Esguince de Tobillo en la Fase Aguda en el Primer Nivel de Atención. (2013)
9. Lisinski, P., Huber, J., Samborski, W., Witkowska, A.: Neurophysiological assessment of the electrostimulation procedures used in stroke patients during rehabilitation. *The International journal of artificial organs*, 31(1), pp. 76–86 (2008)
10. Amiridis, I. G., Arabatzi, F., Violaris, P., Stavropoulos, E., Hatzitaki, V.: Static balance improvement in elderly after dorsiflexors electrostimulation training. *Eur. J. Appl. Physiol.*, vol. 94, no. 4, pp. 424–33 (2005)
11. Skahan, M. S., Lunau, K. R., Combs, D. K., Fischer, W. R.: Electrostimulation System. U.S. Patent Application 14/234,984 (2012)
12. Snyder-Mackler L., Delitto A. Bailey S.L. et al.: Strength of the quadriceps femoris muscle and functional recovery after reconstruction of the anterior cruciate ligament. *J. Bone Joint Surg.*, 77- A(8), pp. 1166–1173 (1995)
13. Lieber R.L., Silva P.D., Daniel D.M.: Equal effectiveness of electrical and volitional strength training for quadriceps femoris muscles after anterior cruciate ligament surgery. *J. Orthop. Res.*, 14(1), pp. 131–138 (1996)
14. Paternostro-Sluga T., Fialka C. Alacamlioglu Y, et al.: Neuromuscular electrical stimulation after anterior cruciate ligament surgery. *Clin. Orthop. Rel. Res.*, Vol. 368, pp. 166–175 (1999)

Modelado difuso Takagi-Sugeno para sintonizar un controlador por calefacción en un invernadero

Julio C. Ramos-Fernández¹, Marco A. Márquez-Vera¹, Luis F. Cerecero-Natale¹,
F. Lafont^{2,3}, J. F. Balmat^{2,3}

¹ Departamento de mecatrónica, Universidad Politécnica de Pachuca,
Zempoala, Hgo., México

² Aix Marseille Université, CNRS, ENSAM, LSIS UMR 7296, 13397 Marseille,
France

³ Université de Toulon, CNRS, LSIS UMR 7296, 83597 La Garde, France

{jramos, marquez, luisfidelmecha}@upp.edu.mx
{lafont, balmat}@univ-tln.fr

Resumen. Este artículo trata sobre el modelado y el control de temperatura en un invernadero, con el fin de compensar las perturbaciones ante la presencia de heladas. Las condiciones de operación involucran una ventilación pasiva, y la irrigación se encuentra apagada. De manera que sólo se controla la temperatura mediante calefacción. Para sintonizar el controlador, un modelo difuso Takagi-Sugeno fue empleado y contruido usando el algoritmo fuzzy c-means para las premisas y mínimos cuadrados para los consecuentes. Mediante mediciones experimentales, donde se activó la calefacción, fue posible sintonizar un modelo difuso. Cada regla difusa corresponde a un submodelo, esta técnica es conocida como compensación paralela distribuida (CPD). El controlador propuesto está basado en el lugar geométrico de las raíces para cada submodelo lineal. La adquisición de datos del microclima se realizó LabVIEW. Los resultados obtenidos son para aplicarse a una válvula proporcional para calefacción, y se muestran los resultados en simulación para dos tipos de aprendizaje: local y global para los consecuentes de los submodelos difusos del tipo Takagi-Sugeno.

Palabras clave: control por calefacción, modelo difuso, invernadero.

1. Introducción

Ante el crecimiento demográfico de la población humana, es necesario producir alimentos en grandes cantidades, para mantener las condiciones climáticas requeridas para un cultivo se emplean invernaderos, el control climático dentro del invernadero es importante para la calidad y la cantidad esperada de la producción agrícola. La temperatura es una variable climática importante para el crecimiento de las plantas [1], esta variable puede ayudar a los procesos fisiológicos de las plantas, pero fuera de determinados rangos, la temperatura puede causar un daño irreversible en el cultivo. Con la producción en invernaderos se protegen las plantas de plagas y se mantiene la

humedad al interior, actualmente se trabaja en los cultivos mediante invernaderos [2]. En la bibliografía en relación a la agricultura protegida, el modelado y control se enfoca principalmente en el clima y la irrigación.

En [3] se realiza identificación difusa Takagi-Sugeno (T-S) mediante clustering para un invernadero, en este tipo de sistemas difusos, los consecuentes de las reglas son combinaciones lineales de las entradas, también se realiza una aproximación multimodelo con resultados satisfactorios, aunque no se aplica un control climático.

En [4] se propone un modelo difuso para modelado y la identificación paramétrica del clima de invernaderos, muestran una estructura jerárquica en la base de reglas del modelo difuso, con esta idea se puede reducir el número de reglas difusas considerando un modelo global del clima del invernadero es aplicado usando un balance de energía. La aproximación encontrada de las variables climáticas es muy similar a las variables reales, sin embargo, en este trabajo se requiere una gran cantidad de mediciones para validar el modelo. Además, la identificación paramétrica es usada en un mecanismo adaptable en línea.

Para mantener un clima adecuado dentro de un invernadero, en el presente trabajo se empleó un sistema difuso T-S para identificar un multimodelo para la temperatura, y consecuentemente proponer una estrategia de control en base al modelo que se propone, Para cada regla difusa se propone un controlador en particular del tipo clásico o lineal [5].

De esta manera, es necesario tener mediciones de diferentes variables climáticas, a fin de tomar en cuenta las perturbaciones presentes del medio ambiente debido a su interrelación no lineal.

Se han desarrollado algunos modelos en la literatura, de manera que se tomó en cuenta una estructura clásica mostrada en [6], en este trabajo se usaron tres funciones de pertenencia para la partición difusa, de manera que se tendrían 3^6 reglas difusas, una estructura jerárquica emplea submodelos colaborativos, e.g. usando tres submodelos para el mismo sistema, es posible usar sólo $3(3^2)$ reglas, este modelo es más fácil de implementar.

En [7, 8] y se emplean mínimos cuadrados para obtener los consecuentes de las reglas, en el presente estudio se usará esta técnica siendo muy similar a la mostrada también en [9]. No sólo existen modelo difusos con la aplicación a invernaderos, en [10] se presenta un modelo basado en algoritmos evolutivos. Una aplicación de control con redes neuronales artificiales se presenta en [11], donde se usa un clasificador ART2.

Para controlar el microclima al interior de un invernadero, lo más común es regular la temperatura interna dentro de un rango recomendado para el cultivo a tratar, en [12] se presenta un control difuso implementado en un FPGA, el control difuso puede ser optimizado como se muestra en [13]. La regulación de la temperatura es importante durante la noche, cuando la temperatura puede descender por debajo de valores críticos, en [14] se presenta un control difuso T-S mediante aire caliente. La humedad relativa al interior es otra variable importante a tratar, en [15] se realiza un control de humedad, para minimizar el consumo de agua se utiliza un control predictivo, un control de humedad para invernaderos se muestra en [16].

El trabajo que aquí se muestra, está organizado de la siguiente manera, primero, en la Sección 2 se describe el invernadero experimental, la subsección 2.2 presenta el modelo difuso, el diseño del controlador se muestra en la subsección 2.3. Los

resultados del control de temperatura, así como la aplicación de únicamente tener un control saturado mediante calefacción son dados en la Sección 3. Finalmente, las conclusiones están en la Sección 4.

2. Materiales y métodos

Para tener una idea sobre el desempeño del control difuso, un modelo puede ser útil en simulación, donde es posible apreciar el desempeño del controlador en lazo cerrado, en esta sección se describe el proceso para obtener un modelo difuso de la temperatura al interior de invernadero, a partir de este modelo se diseña el controlador con la metodología PDC, la cual consiste en diseñar diferentes controladores que actuarán al mismo tiempo, i.e. en paralelo, esto será posible al tener tres submodelos lineales y se diseña un control para cada submodelo, realizando la unión de ellos de manera difusa. Por último, se muestra la simulación así como los criterios de error para tener una noción del desempeño del controlador para este tipo de sistemas.

2.1. Descripción del invernadero

El invernadero se encuentra en la ciudad de Pachuca de Soto, Hidalgo, las características del mismo se presentan en la Tabla 1. El sistema de calefacción es por aire caliente forzado, en esta aplicación una servoválvula fue desarrollada para regular el flujo de gas. Por eso es llamado control por calefacción proporcional, en contraste con un sistema tipo (on-off) donde se contaría con una modulación por ancho de pulso (PWM).

Una importante contribución en este trabajo es mostrar la forma de sintonizar un controlador PDC con una estructura proporcional + integral (PI) difusa, implementado para una servoválvula con respuesta proporcional para regular el flujo de gas de la ignición.

2.2. Modelo difuso

Trabajar con un sistema que cuenta con plantas vivas hace difícil el encontrar un modelo analítico, ya que se trabaja con dinámicas no bien conocidas y variantes en el tiempo, afortunadamente, la lógica difusa emergió como una alternativa para trabajar con información imprecisa o incompleta, así como para tomar decisiones sobre procesos con incertidumbres no aleatorias, de manera similar a como lo hace el lenguaje natural [17]. La noción de conjunto difuso provee un conveniente punto de vista para construir un sistema de inferencia, pero en una forma más general que con conjuntos clásicos [18]. En la lógica difusa, la idea principal es trabajar con variables lingüísticas más fáciles de entender en lugar de usar variables numéricas.

Un sistema difuso es básicamente un conjunto de reglas, las reglas difusas son procesadas por un sistema de inferencia que las evalúa para llegar a una conclusión, puede ser visto como una interpolación entre diferentes subsistemas. Las reglas difusas determinan qué acción hacer mediante variables lingüísticas, las cuales se definen mediante funciones de pertenencia, el conjunto de estas funciones determina

la partición difusa del universo de discurso. En [19] se propone una forma de hacer automáticamente la partición difusa.

Tabla 1. Especificaciones del invernadero.

Parámetro	Características
Coordenadas	Lat: 20.1225° N, Long: -98.7358 O
Ventilación	Pasiva con cuatro cortinas de nylon
Estructura	Galvanizada
Volumen	2270m ³
Túneles	Cuatro túneles de 125m ² con altura de 4.54m
Calefacción	Aire forzado calentado por combustión de gas
Sensor de temperatura	LM35 lineal
Sensor de humedad	HIH4000
Sensor de radiación solar	Radioresistencia
Sistema de adquisición de datos	PIC con ADCs a 10 bits
Sistema de comunicación	Comunicación serial por RF con X-bee
Software	LabVIEW
Cultivo	Jitomate
Plantas por m ²	Ocho
Irrigación	Por sistema gota a gota

Son dos los sistemas de inferencia difusos los más usados, el tipo Mamdani y el T-S [17], en el primero, los consecuentes de las reglas son difusos, mientras que en el T-S los consecuentes son combinaciones lineales de las entradas llegando a ser incluso constantes.

En el presente trabajo, el sistema de inferencia es un tipo T-S lo que a su vez simplifica el paso de quitar la parte difusa al agregar los consecuentes de las reglas [18], el sistema de inferencia describe la temperatura interna dentro de ciertos valores.

Una representación para las reglas difusas se muestra en (1)

$$R^i : \text{if } x(k) \text{ is } v^i, \text{ then } \tilde{T}_i(k+1) = \theta^i x(k), \quad (1)$$

donde R^i es la i -ésima regla, $x(k)$ es un vector que contiene las mediciones incluyendo: la temperatura tanto interna como externa, la humedad relativa, la velocidad del viento, radiación solar, déficit de presión de vapor, ventilación pasiva y el aire forzado caliente, que es la entrada de control, de manera que el vector de estado es $x(k) = [T_i(k), RH_i(k), T_e(k), Rad(k), VPD(k), Cal(k), Ven(k)]^T$, v_i representa la i -ésima función de pertenencia, la cual puede representar el centro de un cluster en la i -ésima regla o submodelo, θ_i es el vector de consecuentes, $\tilde{T}_i(k+1)$ es la aproximación de la temperatura interna, la cual se puede ver como la interpolación entre modelos que son una combinación lineal de las estradas. Para determinar los consecuentes, se emplearon mínimos cuadrados [7, 20].

Se emplearon variables climáticas para obtener un modelo difuso: la temperatura y la humedad relativa internas son $T_i(k), H_i(k)$, la temperatura externa es $T_e(k)$, la

radiación solar se representa con $Rad(k)$, el déficit de presión de vapor se denota por $VPD(k)$, la variable manipulada es $Cal(k)$ y la ventilación pasiva es $Ven(k)$.

Se utilizó el producto como T-norma para realizar la intersección entre conjuntos difusos, la agregación de la reglas fue usando el máximo como operador de unión, la salida del sistema difuso puede calcularse con (2)

$$\tilde{T}_i^j(k+1) = \frac{\prod_{i=1}^c \mu_{\tilde{A}_i} v_i(k)}{\sum_{j=1}^n \prod_{i=1}^c \mu_{\tilde{A}_i} v_i(k)} \theta_j, \quad (2)$$

donde j representa la j -ésima regla, i denota el i -ésimo conjunto difuso, c es el número de conjuntos difusos, n es el número de reglas difusas, θ_j es el consecuente de la regla, $v_i(k)$ es la señal de entrada. Aquí, $\mu_{\tilde{A}_i}$ representa el valor de pertenencia al conjunto difuso \tilde{A}_i .

Una vez definidas las entradas y los operadores difusos, el algoritmo de clustering es aplicado para obtener los conjuntos difusos que mejor describan el comportamiento de la temperatura interna, el algoritmo empleado fue el fuzzy-c means [7]; la interrogante podría ser cuántos clusters deberían utilizarse, pocos clusters presentarían un desempeño pobre, y muchos pueden sobremodelar al sistema. Ya que se ha obtenido un modelo, es necesario validarlo debido a que ante los valores empleados para obtenerlo se puede presentar un buen desempeño y no así con otros valores de entrada.

En este trabajo se emplearon tres clusters para identificar la temperatura interna, el exponente usado por el fuzzy-c means fue $m = 2.5$, el valor de tolerancia fue $\varepsilon = 0.001$. Los datos usados para obtener el modelo fueron tomados el 24 de mayo de 2014. Se tomaron 423 muestras entre las 12:00 am y las 08:00 pm para cada variable climática, para la validación del modelo se tomaron 394 muestras alrededor de las horas mencionadas pero un día después, el periodo de muestreo fue de 2 minutos.

Fueron aproximados dos modelos difusos, el primero empleando una aproximación local; para este caso, se minimizó la función determinada por (3)

$$J_L = \frac{1}{2} \sum_{i=1}^c \sum_{j=1}^n f_i(k) (T_j(k) - \tilde{T}_j(k))^2, \quad (3)$$

donde $f_i(k)$ es una función de validación, y $\tilde{T}_i(k) = v^T(k-1)\theta_i$ es el modelo difuso de la temperatura interna. Si no se emplea la función de validación, la función de costo daría un aprendizaje para un modelo global [21], ahora se obtiene la matriz extendida Φ_g de la siguiente manera:

$$\Phi_g = [W_1\Phi \ W_2\Phi \ \dots \ W_n\Phi],$$

donde la matriz cuadrada $W_i(k) = \text{diag}[\mu_i(k)]_{k=1}^n$ tiene los valores de pertenencia en la diagonal, los consecuentes para las reglas se encuentran usando (4)

$$\tilde{\theta} = (\Phi_g^T \Phi_g)^{-1} \Phi_g^T T_i. \quad (4)$$

Las simulaciones de los modelos aproximados se muestran en la Fig. 1, empleando el algoritmo fuzzy-c means se encontró el modelo local, el otro modelo fue obtenido con los consecuentes obtenido al utilizar (4) [21].

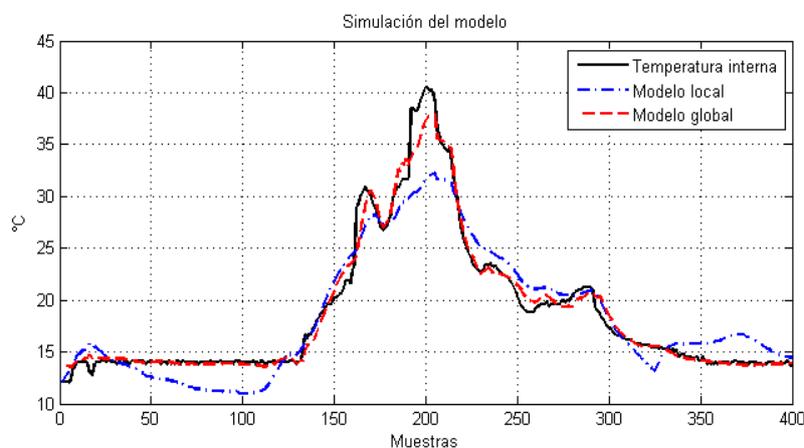


Fig. 1. Simulación del modelo.

Para validar el modelo, se realizaron mediciones de las variables climáticas el día 25 de mayo de 2014 y la temperatura arrojada por el modelo se muestra en la Fig. 2. Los errores medios y sus variaciones se presentan en la Tabla 2.

Tabla 2. Error en el modelo difuso.

Parámetro	Error medio (°C)	Variación del error (°C)
Modelo local	1.9834	6.9600
Modelo global	0.8888	1.5903

2.2. Diseño del controlador

La acción de control fue desarrollada usando el lugar geométrico de las raíces [22], al emplearse tres clusters se obtuvieron tres submodelos lineales dados por los consecuentes de las reglas difusas con la siguiente estructura

$$G_{\text{mod}_i}(z) = \frac{\theta_{\gamma_i}}{z - \theta_{\lambda_i}}$$

donde θ_{ii} representa el primer elemento del vector de consecuentes para el i -ésimo modelo. Dado que el periodo de muestreo fue de 2 min, se propuso tener un tiempo de establecimiento para el sistema de 10 min, así como tener 10 muestras por cada ciclo de oscilación del sistema, en tiempo discreto se tiene que $z = e^{Ts}$, de manera que $|z| = e^{T\sigma}$ y $\angle z = 2\pi\omega_d / \omega_s$, siendo $\sigma = \text{Re}(s)$ y $\omega_d = \text{Im}(s)$.

De esta manera, del método de clustering, se propone que los polos deseados para cada submodelo sean

$$z = 0.4425 + 0.0780i,$$

como cada submodelo encontrado es de primer orden, la función de transferencia del controlador para cada modelo está dada por (5)

$$G_{Di}(z) = \frac{K_i(z + \beta_i)}{(z - 1)(z + \alpha_i)}, \quad (5)$$

donde $\beta_i = -\theta_{ii}$ es un cero que cancela al polo del submodelo determinado por el primer elemento del vector de consecuentes del i -ésimo cluster, el polo en $z = 1$ es para tener un integrador con el fin de eliminar el error en estado estable o estacionario.

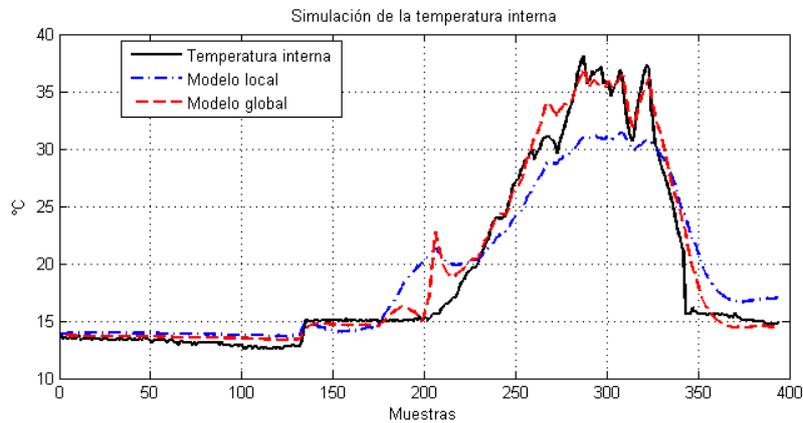


Fig. 2. Validación de los modelos.

Para colocar los polos en la ubicación deseada, es necesario que la suma de los ángulos de los polos y ceros hacia la ubicación deseada sea de 180° , el ángulo ϕ_i requerido se obtiene con (6)

$$\phi_i = 180^\circ - 90^\circ + \arctan \frac{1 - \text{Re}(z)}{\text{Im}(z)}, \quad (6)$$

siendo $\phi_i = 7.9646^\circ$ el polo necesario para tener este ángulo se obtiene con (7)

$$\alpha_i = \frac{\text{Im}(z)}{\tan(\phi_i) - \text{Re}(z)}. \quad (7)$$

Finalmente, para encontrar la ganancia de control K_i , se toma en cuenta que la magnitud del sistema en tiempo discreto debe ser 1, de manera que la ganancia se determina con (8)

$$k = \left\| \frac{(z-1)(z+\alpha_i)}{\theta_{7i}} \right\|_{z=0.4425+0.0780i}, \quad (8)$$

donde θ_{7i} representa el 7-ésimo elemento del vector de consecuentes del i -ésimo cluster, una descripción gráfica del diseño del controlador se presenta en la Fig. 3.

Estos controles locales son definidos para cada regla difusa con el fin de encontrar la señal de control que se aplicará en el sistema, las reglas del control difuso están dadas por (9).

$$R^i : \text{if } x(k) \text{ is } v^i, \text{ then } Cal(k) = K_i e(k-1) + K_i \beta_i e(k-2) + (1-\alpha_i) Cal(k-1) - \alpha_i Cal(k-2). \quad (9)$$

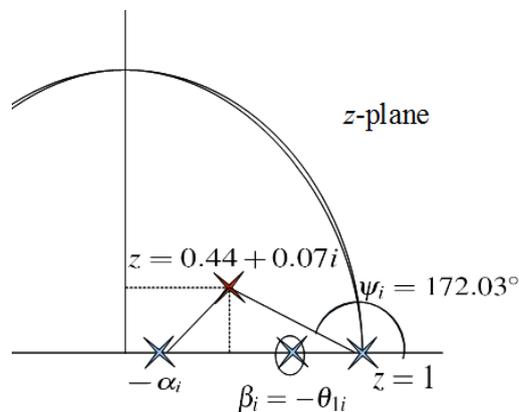


Fig. 3. Diseño del controlador.

El esquema del controlador se muestra en la Fig. 4, donde los submodelos dan los parámetros usados para los controles locales, las variables $T_i(k-1)$ y $H_i(k-1)$ son usadas para obtener el $VPD(k-1)$, este modelo es usado para las simulaciones.

En la Fig. 5 es posible ver la simulación del control usando ambos modelos. En estas simulaciones no existe una restricción para la señal de control y los valores entre las muestras 200 y 350 pueden ser vistas como el resultado de tener perturbaciones en un sistema de primer orden, las cuales hace que la referencia varía alrededor del valor

de referencia, en la práctica es posible que se tenga una saturación en la señal de control, pero en este caso no se logró el rechazo de las perturbaciones sólo con la acción integral del controlador.

3. Resultados

En el invernadero experimental, la acción de control es dada por un calefactor, de manera que la señal sólo puede dar valores positivos entre 0 y 100%, i.e. no se cuenta con un sistema de enfriamiento. Tomando en cuenta esta situación, en la Fig. 6 se muestra el control para una señal solo con el signo positivo, aquí es posible ver que cuando la temperatura interna es mayor que la referencia, la simulación llega a los valores de temperatura dados por el modelo sin acción de control, el cual fue obtenido de con el modelo difuso y las mediciones experimentales realizadas previamente, tal como se muestra en la Fig. 2, lo cual ocurren entre las muestras 200 y 350.

Los errores medios y sus variaciones para el esquema de control mostrado se aprecian en la Tabla 3. Al saturar la señal de control, no es posible apreciar una diferencia importante entre usar un modelo local o uno global, con los valores tomados de la Fig. 5 se calculó la integral del error cuadrático (ISE) (10) y la integral del absoluto del error (IAE) (11) [23] para tener una mejor idea sobre el desempeño del controlador difuso. Los resultados de estos criterios de error están en la Tabla 4, donde es posible ver que el sobre-modelado de usar un aprendizaje global produce alrededor del doble de error comparado con un aprendizaje local.

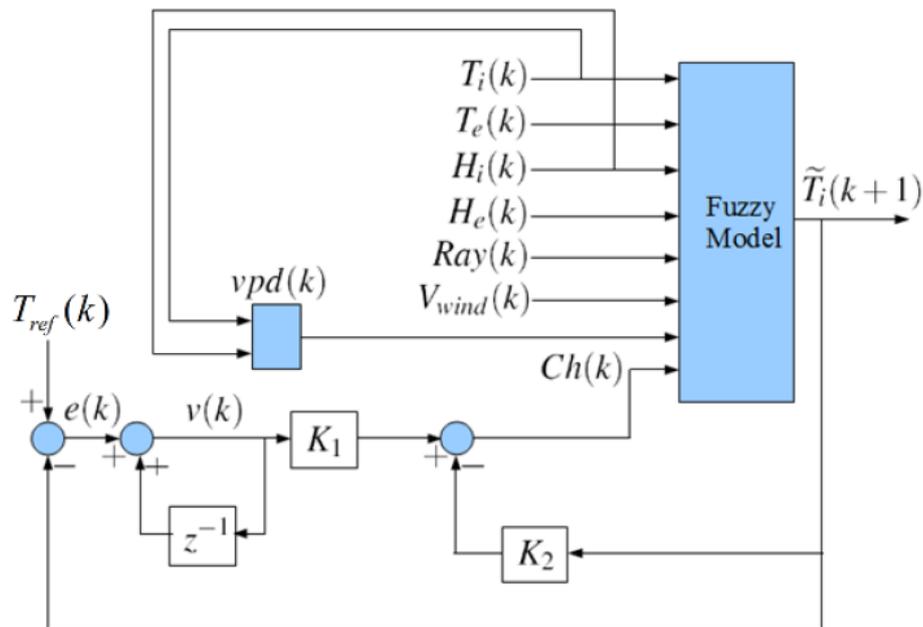


Fig. 4. Esquema del controlador difuso.

$$ISE = \int_0^T e^2(t)dt, \quad (10)$$

$$IAE = \int_0^T |e(t)|dt, \quad (11)$$

Tabla 3. Error del controlador con aprendizaje local y global.

Parámetro	Error medio °C	Variación del error °C
Modelo local	1.4606	5.3520
Modelo global	2.2771	15.6989

Tabla 4. Comparación con diferentes criterios de error.

Método	ISE	IAE
Modelo local	16,562.4720	5,946.2657
Modelo global	32,718.2758	10,044.0510

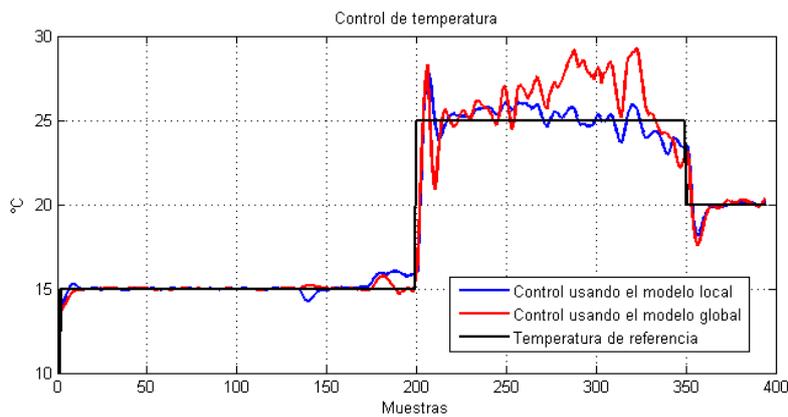


Fig. 5. Simulación del control de temperatura.

4. Conclusiones

El modelado difuso es una alternativa cuando es difícil obtener un modelo matemático, también es posible usar el conocimiento de un experto en el sistema a modelar, esto se puede plasmar en un modelo difuso, en este trabajo se usaron dos técnicas para obtener un modelo. El usar un aprendizaje local funciona como un filtro eliminando las variaciones en la temperatura, el aprendizaje global mostró un mejor desempeño en la aproximación en comparación con el aprendizaje local, pero al tener un modelo más parecido al comportamiento real del sistema puede causar un sobre-modelado al tomar en cuenta las perturbaciones y el ruido o errores en la medición.

El control difuso aplicado a ambos modelos es similar a un PI, ya que se tiene un polo en $z = 1$, para contar con los polos deseados, fue agregado un polo más con lo que se cumple la condición de hacer 180° a la ubicación deseada, la ganancia del

control es necesaria para de mantener los polos en un lugar específico dentro del lugar geométrico de las raíces. Un control en particular fue diseñado para cada regla difusa, siendo tres en este caso, estas acciones de control están enlazadas de forma difusa para obtener la señal de control a aplicar al sistema.

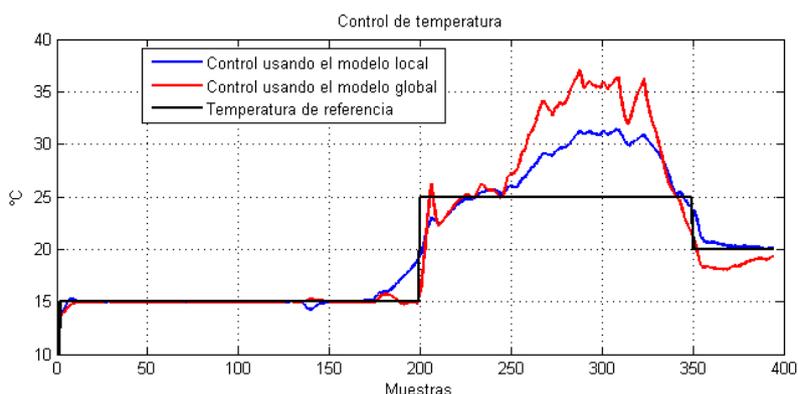


Fig. 6. Control de temperatura sólo con calefacción.

En simulación fue posible ver cómo el diseño del controlador que usa un modelo con aprendizaje local mostró un mejor desempeño que con aprendizaje global, en base a los criterios de error utilizados, dado que en el sistema real no es posible enfriar el invernadero cuando el error de control de la calefacción se hace positivo, en la Fig. 6 se mostró la simulación de saturar el controlador para tener únicamente calefacción, aquí se puede ver cómo entre las muestras 200 y 350, cuando la temperatura interna es superior a la referencia, el sistema sigue los modelos difusos según sea el tipo de aprendizaje usado para modelar el clima. Por otra parte, el control usa los parámetros obtenidos por el modelo difuso con los aprendizajes local o global para regular la temperatura usando el modelo respectivo.

Como trabajo futuro se propone contar con un sistema de enfriamiento para el invernadero, y así aplicar el sistema de control en lazo cerrado. Finalmente un control con modelo interno ayudará al rechazo de perturbaciones, siendo en este caso, la velocidad del viento, la radiación solar y la humedad relativa dentro y fuera del invernadero.

Referencias

1. Vermeulen, K., Aerts, J. M., Dekock, J., Bleyaert, P., Berckmans, D., Steppe, K.: Automated leaf temperature monitoring of glasshouse tomato plants by using a leaf energy balance model, *Computers and Electronics in Agriculture*. 87 (2012) 19–31
2. Bennis, N., Duplaix, J., Enea, G., Haloua, M., Youlal, H.: Greenhouse climate modelling and robust control, *Computers and Electronics in Agriculture*. 61 (2008) 96–107
3. Trabelsi, A., Lafont, F., Kamoun, M., Enea, G.: Fuzzy identification of a greenhouse, *Applied Soft Computing*. 7 (2007) 1092–1101

4. Salgado, P., Cunha, J. B.: Greenhouse climate hierarchical fuzzy modelling, *Control Engineering Practice*. 13 (2005) 613–628
5. Wang, H. O., Tanaka, K., Griffin, M. F.: Parallel distributed compensation of nonlinear systems by takagi-sugeno fuzzy model, *Proc. FUZZY-IEEE/IFES* 95(1) (1995) 531–538
6. Márquez, M. A., Waissman, J., Gutiérrez, O.: Fuzzy model based iterative learning control in phenol biodegradation, *Foundations of Fuzzy Logic and Soft Computing*. 4529 (2007) 328–337
7. Babuška, R.: *Fuzzy Modeling for Control*, International Series in Intelligent Technologies, Aachen, Germany (1998)
8. Abdel-Rahman, E. M., Mutanga, O., Odindi, J., Adam, E., Odindo, A., Ismail, R.: A comparison of partial least squares (pls) and sparse pls regressions for predicting yield of swiss chard grown under different irrigation water sources using hyperspectral data, *Computers and Electronics in Agriculture*. 106 (2014) 11–19
9. Gorrostieta-Hurtado, E., Sotomayor-Olmedo, A., Pedraza-Ortega, J. C., Aceves-Fernandez, M. A., Villaseñor-Carillo, U. G.: Modeling key parameters for greenhouse using fuzzy clustering technique, *Proc. Conf. of Ninth Mexican International Conference on Artificial Intelligence*. 9 (2010) 103–106
10. Herrero, J., Blasco, X., Martínez, M., Ramos, C., Sanchis, J.: Non-linear robust identification of a greenhouse model using multi-objective evolutionary algorithms, *Biosystems Engineering*. 98 (2007) 335–346
11. Fourati, F.: Multiple neural control of a greenhouse, *Neurocomputing*. 139 (2014) 138–144
12. Castañeda-Miranda, R., Ventura-Ramos, E., del Río Peniche-Vera, R., Herrera-Ruiz, G.: Fuzzy greenhouse climate control system based on a field programmable gate array, *Biosystems Engineering*. 94 (2006) 165–177
13. Lafont, F., Balmat, J. F.: Optimized fuzzy control of a greenhouse, *Fuzzy Sets and Systems*. 128 (2002) 47–59
14. Nachidi, M., Rodriguez, F., Tadeo, F., Guzman, J.: Takagi-sugeno control of nocturnal temperature in greenhouses using air heating, *ISA Transactions*. 50 (2011) 315–320
15. Körner, O., Challa, H.: Process-based humidity control regime for greenhouse crops, *Computers and Electronics in Agriculture*. 39 (2003) 173–192
16. Blasco, X., Martínez, M., Herrero, J. M., Ramos, C., Sanchis, J.: Model based predictive control of greenhouse climate for reducing energy and water consumption, *Computers and Electronics in Agriculture*. 55 (2007) 49–70
17. Ross, T.: *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, Ltd., West Sussex (2008)
18. Passino, K., Yurkovich, S.: *Fuzzy Control*, Addison Wesley Longman Inc., California (1998)
19. Su, Z. G., Wang, P. H., Shen, J., Li, Y. G., Zhang, Y. F., Hu, E. J.: Automatic fuzzy partitioning approach using variable string length artificial bee colony (vabc) algorithm, *Applied Soft Computing*. 12 (2012) 3421–3441
20. Sabzevar, M., Naghibzadeh, M.: Fuzzy c-means improvement using relaxed constraints support vector machines, *Applied Soft Computing*. 13 (2013) 881–890
21. Foulloy, L., Galichet, S., Titli, A.: *Commande floue 2: de l'approximation à l'apprentissage*, volume 2, Hermes Science Publications, France (2003)
22. Kuo, B.: *Sistemas de Control Digital*, Compañía Editorial Continental, México, (2003)
23. Åström, K., Hägglund, T.: *Advanced PID Control*, Instrumentation, Systems, and Automation Society, United States of America (2006)

Monitoreo de emociones aplicadas a terapias basadas en juegos y lógica difusa para adultos mayores

Monica Janneth Guido Silva, Víctor Manuel Zamudio Rodríguez,
Héctor José Puga Soberanes, Juan Martín Carpio Valadez,
María del Rosario Baltazar Flores

Instituto Tecnológico de León,
León, Guanajuato, México

m142403@itleon.edu.mx

Resumen. La cantidad de personas de edad adulta en México ha aumentado gradualmente, esto ha sido posible gracias al desarrollo médico y tecnológico, lo cual en definitiva es un logro de alto impacto social. Sin embargo con este aspecto, se tiene también un aumento de personas que debido a su edad son más susceptibles a padecer algún tipo de demencia, como el Alzheimer. Investigaciones recientes están tomando planes de acción y medidas preventivas, ejemplo de ello son las terapias no farmacológicas, como las terapias basadas en juegos del sistema “Mente Activa”, el cual es parte de una investigación previa más general y fue desarrollado en el Instituto Tecnológico de León en colaboración con el Instituto de la Memoria, para la detección y prevención del deterioro cognitivo, dicho sistema emplea técnicas de lógica difusa y clasificación. Y genera planes de estimulación cognitiva al evaluar los parámetros como la escolaridad del usuario y los resultados de la prueba neuropsicológica Neuropsi, entre otros. Para mejorar la generación de planes de estimulación, se busca integrar un módulo para el monitoreo de emociones mediante el reconocimiento de expresiones faciales, utilizando el método de base característica, el cual consiste en aplicar técnicas de visión artificial y la extracción de puntos característicos de las regiones faciales de interés. En este trabajo se muestra el reconocimiento de expresiones faciales asociadas a la felicidad, tristeza, sorpresa y un estado normal con personas de edad adulta. Los resultados de la primera etapa del proyecto se muestran alentadores. Posteriormente se añadirá la variable de emoción al sistema de lógica difusa del sistema “Mente Activa”.

Palabras clave: Alzheimer, cómputo afectivo, demencia, deterioro cognitivo, lógica difusa, reconocimiento de expresiones faciales, terapia asistida por computadora, terapias basadas en juegos.

1. Introducción

La población mexicana está experimentando cambios socio-demográficos, como el aumento de la cantidad de personas de mayor edad y el descenso paulatino en el índi-

ce de mortalidad, de acuerdo con datos del INEGI (Instituto Nacional de Estadística, Geografía e Informática) [1], esto ha sido posible debido a la expansión y mejoras en los servicios de salud, así como al desarrollo en el campo médico y tecnológico, lo cual es un logro social importante, particularmente en México se estima que el porcentaje de adultos mayores incrementa a 12.5 por ciento en 2020 y 28.0 por ciento en 2050, para adquirir un perfil propio de las poblaciones envejecidas, de acuerdo con el INAPAM (Instituto Nacional de las Personas Adultas Mayores) [2].

Con este aumento de población se tiene también un incremento en las enfermedades que puede afectar a la misma, como los trastornos cognitivos, la depresión, la ansiedad y las enfermedades crónico-degenerativas [2, 3]. Se estima que a nivel global en 2010 había 35.6 millones de personas con demencia, se prevé que para el 2030 esta cifra se duplicará a 65.7 millones y 115.4 millones en 2050 [3]. Por lo cual la salud mental del adulto mayor emerge como una prioridad social.

Con el propósito de prevenir y contrarrestar dichos problemas de salud mental, han surgido novedosas investigaciones, como las terapias no-farmacológicas, tal es el caso de las terapias basadas en juegos del sistema “Mente Activa”, con el cual se puede detectar y prevenir el deterioro cognitivo [4], dicho sistema emplea técnicas de clasificación y de lógica difusa, al evaluar ciertos parámetros (como el grado de escolaridad del usuario y los resultados obtenidos de las pruebas neuropsicológicas; Neuropsi y Minimental), en el módulo inteligente del sistema de “Mente Activa” [5], para generar de manera automática un plan de estimulación individual.

Sin embargo se plantea la cuestión de que los resultados del desempeño puedan verse afectados por el estado emocional del usuario, de acuerdo con el enfoque del Cómputo Afectivo [6]. Por lo que se tiene la propuesta de desarrollar e incorporar un módulo para el monitoreo de emociones, durante la interacción del usuario con los juegos del sistema “Mente Activa”. En esta investigación se presenta una breve descripción de los puntos importantes relacionados con el deterioro cognitivo, el cómputo afectivo, un análisis sobre la tecnología usada para mejorar el desempeño cognitivo y el uso de la lógica difusa en el sistema “Mente Activa”.

También se presenta el proceso para la implementación del módulo de monitoreo de emociones mediante el reconocimiento de expresiones faciales (como sonrisa, tristeza, sorpresa y un estado normal), aplicando el método de base característica de Gurpreet [7] (el cual aplica técnicas de procesamiento de imágenes y visión artificial, para la extracción de puntos característicos de las diferentes regiones faciales, como ojos y boca).

Posteriormente se muestran los resultados de las pruebas de la primera etapa, correspondiente al desarrollo del módulo para el monitoreo de emociones. Para las pruebas se tomaron imágenes, en tiempo real, de 34 participantes de edad adulta, estas imágenes están asociadas a los estados emocionales de sonrisa, tristeza, sorpresa y un estado normal. Los resultados muestran evidencia de la viabilidad del enfoque. Como trabajo futuro se añadirá la variable “emoción” al sistema de lógica difusa de “Mente Activa”, lo cual permitirá mejorar la generación de los planes de estimulación cognitiva individual, fortaleciendo el análisis del desempeño del usuario.

2. Deterioro Cognitivo

Una de las principales discapacidades que se presentan durante la vejez, es la limitación de las habilidades cognitivas.

2.1. Demencia

La demencia es una alteración adquirida persistente y progresiva en la función intelectual, que afecta la memoria y el dominio cognitivo generalmente como: afasia (dificultad para elegir palabras), apraxia (incapacidad para llevar a cabo actividades motoras), agnosia (imposibilidad para identificar objetos), alteración de las funciones de ejecución (mala abstracción, flexibilidad mental, planificación y juicio), la depresión y el delirio también es factible que se presenten con daño cognitivo [8].

2.2. Alzheimer

La enfermedad de Alzheimer es el tipo de demencia más común y generalmente se presenta a partir de los 65 años [9]. Los síntomas más comunes es el deterioro gradual de la habilidad para recordar información reciente. Esto ocurre por el daño en las neuronas, las cuales pueden dejar de funcionar normalmente y mueren, generando cambios en la memoria, el comportamiento y la habilidad de pensar claramente [10].

3. Cómputo afectivo

El término “Cómputo Afectivo” fue empleado por primera vez en [11] por Rosalinda Picard, para describir al cómputo que está relacionado con la influencia de las emociones. El estudio del cómputo afectivo, se ha centrado en el reconocimiento, procesamiento e interpretación y simulación de emociones, con la intención de dotar a la máquina con capacidades de comunicación e interpretación afectiva hacia el usuario, ha tomado un gran auge en la implementación de varios sistemas inteligentes, como de tutoría afectiva, aplicación de reconocimiento de emociones mediante expresiones faciales, uso de dispositivos biométricos así como sistemas con fines terapéuticos y psicológicos.

La idea de aplicar cómputo afectivo en el presente proyecto a través del reconocimiento de expresiones faciales, surge de la siguiente premisa: las decisiones tomadas por el ser humano son motivadas por las emociones [6, 11], lo cual pudiera afectar el desempeño del usuario durante su interacción con los juegos del sistema “Mente Activa”.

3.1. ¿Qué es una emoción?

La definición del concepto “Emoción” es una situación no trivial, debido a que existe una gran gama de definiciones en el estado del arte. Kleinginna muestra en su

artículo un compendio con 92 definiciones [12]. Sin embargo varios investigadores concuerdan en que la emoción es: un conjunto complejo de interacciones entre factores subjetivos y objetivos, mediada por los nervios y la cuestión hormonal [13]. En el presente proyecto se detectará únicamente expresiones faciales relacionadas a los estados emocionales de felicidad, estado normal, sorpresa y tristeza.

4. Tecnología aplicada a la terapia cognitiva

En recientes trabajos se han dado a conocer avances tecnológicos para la terapia cognitiva aplicando cómputo afectivo con distintos propósitos como captar el interés de los usuarios, detectar situaciones en que el usuario manifiesta alguna emoción que pueda afectar el rendimiento de su desempeño o influir en algunos aspectos como nivel de atención, aprendizaje, toma de decisiones, etc. Algunos ejemplos de estas investigaciones son: los sistemas de tutoría afectiva [14], la terapia basada en juegos de Neuroracer [15], el reconocimiento de emociones a través de las expresiones faciales [7] y el uso de dispositivos empleados para análisis de las emociones a través de medidas fisiológicas (como la frecuencia cardíaca y la presión arterial) [16].

5. Sistema inteligente “Mente Activa”

5.1. “Mente Activa”

“Mente Activa” es un programa para la prevención, detección, evaluación y supervisión de los adultos mayores con deterioro cognitivo y la demencia [4]. El sistema permite la estimulación cognitiva a través de la utilización de juegos interactivos especialmente diseñados por psicólogos, el sistema incluye las áreas de estimulación cognitiva como: atención, cálculo, funciones ejecutivas, gnosias, lenguaje, memoria y orientación [4, 5]. Fue desarrollado en el Instituto Tecnológico de León en colaboración con el Instituto de la memoria, como parte de una investigación más general.

5.2. Lógica difusa

La Lógica Difusa es una forma de lógica matemática multivaluada en el que la verdad puede tomar un valor en un rango continuo de valores entre 0 y 1 puesto que utiliza valores lógicos continuos o aproximados entre 0 (completamente falso) y 1 (totalmente cierto). Un enunciado no necesariamente es verdadero o falso, pero puede tener un grado de verdad entre 0 y 1 [19]. Recibe otros dos significados llamados *teoría del razonamiento aproximado* y *la teoría de la lógica lingüística*, la última teoría un tanto marginal, es una de las lógicas cuyos valores de verdad son expresiones de lenguaje natural (por ejemplo, cierto, más o menos cierto, etc.) [17]. Zadeh introdujo el término de Lógica Difusa, determinada como un conjunto de principios

matemáticos para la representación del conocimiento basado en grados de pertenencia [18].

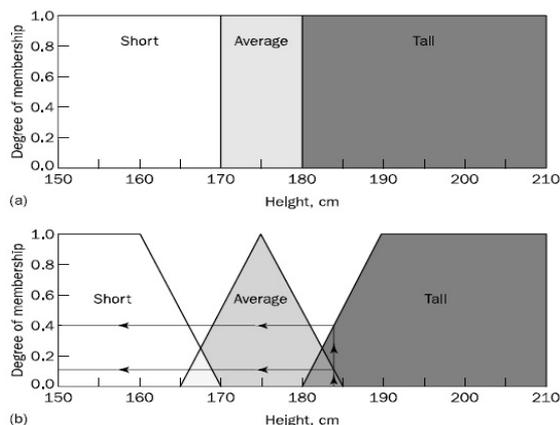


Fig. 1. Esta imagen fue tomada del libro de Inteligencia Artificial de Negnevitsky [19], en este ejemplo de conjuntos difusos se puede apreciar la función de pertenencia donde la variable lingüística altura de un hombre (Height), tiene asociados tres términos lingüísticos (conjuntos difusos): bajo, mediano y alto, para los cuales se tiene asociado un grado de pertenencia, que puede estar entre 0 y 1. En la imagen (a) se pueden ver los conjuntos de manera cruda, en (b) se muestran los conjuntos difusos.

5.3. Sistema de lógica difusa tipo 1 (FLS T1) de “Mente Activa”

Sistema de Lógica Difusa de “Mente Activa” [4] [5], puede ser representado como se muestra en la figura 2. El conocimiento esta incrustado en el conjunto de reglas que alimenta el motor de inferencia del sistema (dadas por los especialistas o psicólogos basándose en antecedentes ó entradas y consecuentes ó salidas difusas). Las entradas en crudo (crisp inputs) consideradas para este sistema son: la escolaridad y la puntuación de los resultados de las pruebas neuro-psicológicas: Neuropsi y Minimal.

Después son fuzzificadas en T1 FSs, que se convierten en nuevos T1 FS por el motor de inferencia. El defuzzificador transforma esta nueva T1 FS en una salida en crudo (crisp output) que es el plan de estimulación cognitiva individual, el cual se pretende mejorar al integrar la nueva variable de entrada “emoción”.

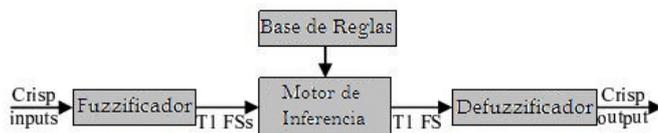


Fig. 2. Esquema de un Sistema de Lógica Difusa tipo 1 de “Mente Activa”, tomado de [5].

6. Proceso de implementación del módulo de monitoreo de emociones (mediante el reconocimiento de expresiones faciales)

La implementación del módulo de monitoreo de emociones está organizada en las siguientes cuatro etapas:

- A) **Detección del rostro:** se utilizó el clasificador “Detect Haar Cascade” de la librería Emgu CV para C#. Emgu Cv es un envoltorio para .NET del conjunto de librerías OPENCV [20], con la cual se puede hacer procesamiento de imágenes y aplicar técnicas de visión artificial. Su aplicación puede verse en Figura 3.



Fig. 3. Ejemplo de la detección del rostro aplicando la librería Emgu CV, la cual se encuentra disponible en [20].

- B) **Reconocimiento de Emociones a través de las expresiones faciales:** se aplicó el método de base característica, para detectar expresiones faciales humanas, el cual puede ser estudiado con mayor detalle en [7, 21]. De manera general el procedimiento que emplea dicho método es el siguiente:

- Pre-procesamiento de la imagen obtenida en la detección del rostro: Se obtiene una imagen RGB (del inglés de *red, green, blue*), la cual es contrastada y transformada en una imagen en blanco y negro, después se realiza la obtención del rostro como área del interés.
- Se buscan el centro de la imagen y se aplica el patrón de seguimiento al desplazarse por cada punto, para detectar los ojos y la boca como nuevas áreas de interés en la imagen.
- Posteriormente se aplican las curvas de Bezier en las áreas de boca y ojos extrayendo los puntos característicos faciales.
- Finalmente la representación de los valores de los puntos obtenidos son comparados en la base de datos que contiene previamente los vectores de puntos asociados con las emociones de sonrisa, tristeza, sorpresa y un estado normal. Por lo que esta clasificación no requiere de un entrenamiento previo.

La técnica de base característica se lleva a cabo detectando cambios característicos en las diferentes regiones faciales, dichas regiones están basadas en el

Sistema de Codificación de la Acción Facial (FACS: Facial Action Coding System). El cual es un sistema basado en la observación humana y permite detectar cambios en las características faciales [7].

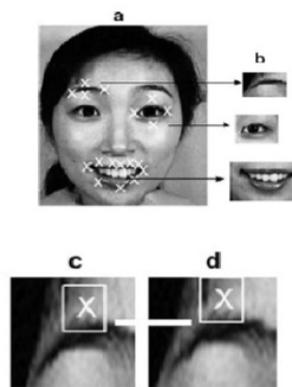


Fig. 4. Imagen, tomada de [21], muestra el proceso para el reconocimiento de emociones a través de las expresiones faciales, aplicando el método de base característica de Gurpreet [7].

Como puede verse en la Fig. 4, “a” Una vez detectado el rostro se procesa la imagen, “b” se extraen las características faciales, “c” y “d” el patrón de seguimiento permite encontrar las regiones de interés. Para el caso particular de esta investigación solo se extrajeron las regiones de interés correspondientes a los ojos derecho e izquierdo y la boca.

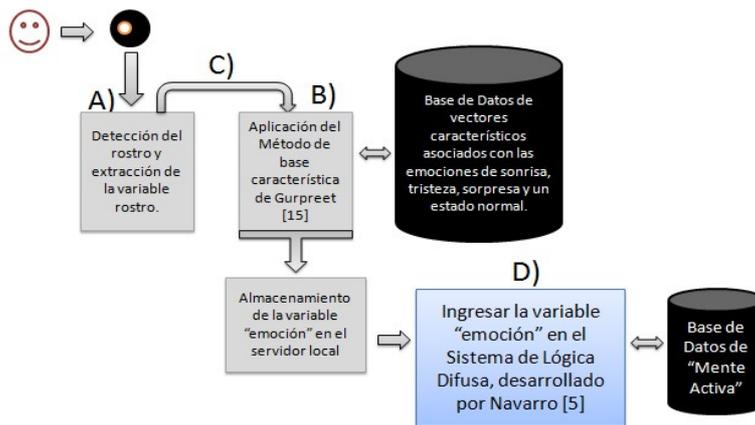


Fig. 5. El status actual de la investigación se encuentra en la etapa de integración de las etapas: a) detección de rostro, b) reconocimiento de emociones a través de expresiones faciales.

- C) **Integración de las primeras dos etapas:** Incisos A y B, en la Figura 5, se muestra el diagrama de integración.
- D) **Ingresar la variable de la emoción detectada en el FLS T1 [4] [5] de “Mente Activa”.** Finalmente la variable “emoción” obtenida será añadida al sistema de lógica difusa de “Mente Activa” como se muestra en el inciso D) de la figura 4, esto permitirá mejorar la generación de planes de estimulación cognitiva individual.

7. Experimentación

Para realizar las pruebas al módulo de monitoreo de emociones mediante el reconocimiento de expresiones faciales, se tomaron 34 imágenes para cada categoría (felicidad, tristeza, sorpresa y normal) a una muestra de 34 participantes de edad adulta, este tamaño de muestra para el experimento es de acuerdo al teorema del límite central [23, 24], para un total de 136 muestras para clasificación. Las imágenes fueron tomadas en tiempo real con la cámara digital de computadora. Para el cálculo de la precisión se empleó la siguiente ecuación, propuesta en [22]:

$$Precision = \frac{+P}{(+P)+FP} \quad (1)$$

donde +P son los verdaderos positivos y FP son los falsos positivos.

A continuación se muestra el porcentaje de precisión del monitoreo de emociones a través de expresiones con imágenes tomadas con la cámara.

Tabla 1. Porcentaje de precisión del reconocimiento de emociones a través de las expresiones faciales con imágenes tomadas de la cámara digital de la computadora con personas de edad adulta. Los números 27, 28, 20 y 28 son los verdaderos positivos de acuerdo con la fórmula de precisión [22].

Categorías	Normal	Feliz	Triste	Sorpresa	Precisión
Normal	27	2	5	0	79.41%
Feliz	2	28	0	4	82.35%
Triste	10	0	20	4	58.82%
Sorpresa	2	4	0	28	82.35%

8. Conclusiones

En este trabajo de investigación se muestra el diseño y desarrollo del módulo para el monitoreo de emociones basado en el estudio del Cómputo Afectivo mediante el

reconocimiento de expresiones faciales, el cual será integrado a un sistema de terapias basadas en juegos y lógica difusa para Adultos Mayores llamado “Mente Activa”.

Para las pruebas realizadas en la primera etapa del módulo se tuvo una muestra de 34 usuarios de edad adulta, tomando un total de cuatro imágenes por persona, cada una correspondiente a las categorías de expresiones faciales asociadas a la felicidad, tristeza, sorpresa y un estado normal, posteriormente dichas imágenes fueron procesadas con el “método de base característica para detectar expresiones faciales humanas” que emplea técnicas de procesamiento de imágenes y de visión artificial.

Los resultados de la experimentación muestran la viabilidad del enfoque, sin embargo las condiciones de experimento son cruciales (como la iluminación, calidad de la imagen, la disposición de participación del usuario, los movimientos bruscos del usuario, entre otras), ya que afectan el índice de clasificación.

El trabajo futuro propuesto es realizar más pruebas con usuarios de edad avanzada y finalmente dicha representación de la variable del estado emocional (variable “emoción”) se añadirá en una segunda etapa donde pueda ser utilizada como una variable de entrada, en el sistema de lógica difusa de “Mente Activa”, y así poder mejorar la generación de planes de estimulación cognitiva y fortalecer el análisis del desempeño del usuario.

Agradecimientos. Agradecimientos Al Consejo Nacional de Ciencia y Tecnología (CONACYT) de México, por el apoyo brindado en esta investigación y a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de León.

Referencias

1. Instituto Nacional de Estadística, Geografía e Informática: Los Adultos Mayores en México; Perfil socio demográfico al inicio del siglo XXI. Aguascalientes, México: INEGI. Disponible en www.inegi.gob.mx, pp. 9–18 (2005)
2. INAPAM: Instituto Nacional de las Personas Adultas Mayores; Gobierno Federal, Ejes reactivos de la política pública nacional a favor de las personas adultas mayores, pp. 13–35, 49 (2010)
3. Gutiérrez Robledo, L. M.: Plan de Acción Alzheimer y Otras Demencias. México 2014, Instituto Nacional de Geriátrica, pp. 19–54 (2014)
4. Navarro, F. J., Zamudio, V., Baltazar, R., Lino, C., Díaz, C., Faiyaz D.: Towards Game Based Monitoring and Cognitive Therapy for Elderly using a Neural-Fuzzy approach. In: Workshops Proceedings in the Mexican International Conference on Computer Science, pp. 1–6 (2013)
5. Navarro, F. J.: Sistema Inteligente para el Monitoreo y Generación de Tareas en Escenarios de Terapia para Adultos Mayores. Instituto Tecnológico de León, León Guanajuato, México, pp. 51–63 (2014)
6. Picard, R. W.: Online Affective computing: challenges. *International Journal of Human Computer Studies*, vol. 59, pp. 55–64, Elsevier Science Ltd., Cambridge, DOI: 10.1016/S1071-5819(03)00052-1 (2003)
7. Gurpreet, S., Vikas, W.: Feature Based Method to detect Human Facial Expressions. *International Journal of Computer Science and Communication Engineering, IJCSCE, India*, pp. 42–47 (2013)

8. McPhee, S. J., Papadakis, M.A.: Diagnóstico clínico y tratamiento. McGraw-Hill, México, pp. 63–65 (2012)
9. Sosa, A. L., Acosta, I.: Epidemiología de las Demencias. Instituto Nacional de Geriatria, México, Sitio web <http://www.geriatria.salud.gob.mx> (2014)
10. Alzheimer's Association: Alzheimer's Disease Facts and Figures. Alzheimer's and Dementia, Volume 10, Issue 2, pp. 5–14 (2014)
11. Picard, R. W.: Affective Computing. MIT, Media Laboratory Perceptual Computing Section Technical Report No. 321, pp. 1–16 (1998)
12. Kleinginna, Jr. P. R., Kleinginna, A. M.: A Categorized List of Emotion Definitions, with Suggestions for a Consensual Definition. *Motivation and Emotion*, vol. 5, no. 4, pp. 345–379 DOI: 0146-7239/81/1200-0345 (1981)
13. Van den Broek, E. L., Janssen, J. H., & Westerink, J. H.: Guidelines for Affective Signal Processing (ASP): From Lab to Life. In: 3rd International Conference on Affective Computing & Intelligent Interaction, Amsterdam, The Netherlands, pp. 704–709 (2009)
14. Mao, X., Li, Z.: Agent based affective tutoring systems: A pilot study. *Computers and Education*, 55, pp. 202–208 (2010)
15. Anguera, J. A., Boccanfuso, J., Rintoul, J. L., Al-Hashimi, O., Faraji, F., Janowich, J. et al.: Video gaming enhances cognitive skills that decline with age: Game Change. Macmillan Publishers, San Francisco, California, pp. 97–10 (2013)
16. Aguiñaga, A. R., López, M., Alanís, A., Baltazar, R. & Zamudio, V. M.: Emotion analysis through physiological measurements. In: J. Botía, & D. Charitos (Ed.), *Workshop Proceedings of the 9th International Conference on Intelligent Environments*, 17, pp. 97–106, IOS Press (2013)
17. Novák, V., Perfilieva, I. and Močkoř, J.: *Mathematical principles of fuzzy logic* Dordrecht. Kluwer Academic (1999)
18. Zadeh, L.: Fuzzy sets. *Information and control*, vol. 8, no. 3, pp. 338–353 (1965)
19. Negnevitsky, M.: *Artificial Intelligence. A Guide to Intelligent Systems*. Pearson Education, pp. 88–93 (2002)
20. EmguCv, Disponible en http://www.emgu.com/wiki/index.php/Main_Page (2015)
21. S. Gurpreet, Baljit, S.: Feature Based Method for Human Facial Emotion Detection using Optical Flow Based Analysis. *An International Journal of Engineering Sciences, Journal Anu Books*, vol. 4 (2011)
22. Fawcett, T.: An Introduction to ROC analysis. *Pattern Recognition Letters*, Elsevier, vol. 27, pp. 861–874 (2006)
23. Grinstead, Ch. M.; Snell, J. Laurie: *Introduction to Probability*. AMS Bookstore, pp. 325–360 (1997)
24. Filmus, Y.: Two Proofs of the Central Limit Theorem. pp. 1–3 (2010)

¿La explicación explica la abducción? Encuentros y desencuentros entre Inteligencia Artificial y Filosofía de la Ciencia

David Gaytán Cabrera

UACM/ Programa *Principhia* / UNAM,
México

Abstract. Dov Gabbay y John Woods defendieron un enfoque no tradicional para entender formalmente la abducción. Una parte significativa de la defensa que proponen Gabbay y Woods está basada en una robusta crítica contra lo que podría llamarse “el enfoque formal tradicional de la abducción”. En este artículo realizaré una evaluación de un punto específico de esta crítica, el cual está relacionado con la noción de explicación. Haré esta evaluación a la luz del importante debate que sobre la caracterización de la explicación científica ha tomado ya más de 5 décadas de discusión fructífera en el ámbito de la Filosofía. Finalmente propondré que hay una falla importante en la crítica realizada por estos autores y que tal falla nos lleva a revalorar la importancia de la interacción entre dos disciplinas de investigación involucradas en estos enfoques: la Inteligencia Artificial y la Filosofía de la Ciencia.

Palabras clave: abducción, explicación, Inteligencia Artificial, Filosofía de la Ciencia.

1. Introducción

En un artículo reciente, Dov Gabbay y John Woods defendieron un enfoque no tradicional para entender formalmente la abducción. Este nuevo enfoque tiene virtudes considerables: puede decirse que se trata de una apuesta por una representación más pluralista y en cierto sentido debilitada de la noción de inferencia abductiva. Algunos rasgos importantes de este nuevo enfoque son, por ejemplo, la inclusión más franca de parámetros para la representación de relevancia, la integración de varios parámetros provenientes de diversas lógicas no clásicas y la distinción entre explicación y abducción. Una parte significativa de la defensa que proponen sus autores está basada en una robusta crítica contra lo que podría llamarse “el enfoque formal tradicional”. Uno de los ítems que constituyen la crítica del artículo se refiere a la perspectiva explicacionista en que según Gabbay y Woods, en el punto de vista tradicional es interpretado el carácter consecuencialista de la abducción. Realizaré una evaluación de este punto específico de crítica relacionado con la explicación, a la luz del importante debate que sobre la caracterización de la explicación científica ha tomado ya más de 5 décadas de discusión fructífera en el ámbito de la Filosofía de la Ciencia. El examen me llevará a mostrar una falla que considero importante en el punto de crítica mencionado. Lo interesante es

que parece ser una falla que penetra la forma en que es tratado el tema desde la Inteligencia Artificial. A partir de lo anterior, intentaré extraer algunas conclusiones finales acerca de la relación entre esta disciplina de investigación y la Filosofía de la Ciencia.

2. El enfoque formal tradicional para la elucidación formal de la abducción, según Gabbay y Woods

Las principales contribuciones originales del enfoque tradicional de la abducción descrito por Gabbay y Woods en [1], en su párrafo 12.1, se deben a varios autores, como Aliseda [2], Kowaliski [3], Kuipers [4], Magnani [5] y, conjuntamente, a Kakas, Kowalski y Mancarella [6]. Gabbay y Woods proponen llamar a la parte nuclear de este enfoque “el esquema *AKM*”, y sostienen que puede describirse simplícidamente mediante los siguientes 8 ítems (incluyo aquí el inciso adicional 1 que se agregó en [1] con el objetivo de facilitar en ese artículo la discusión que ellos desarrollaron alrededor de los requerimientos para toda elucidación de la noción abducción):

1. T!
Que puede leerse como: Se tiene el objetivo de encontrar una explicación para E.
2. E
Que puede leerse como: Se tiene el dato de que el suceso E ocurrió.
3. No ocurre que: $K \Rightarrow E$
Que puede leerse como: K, nuestra base de conocimientos, no tiene a E como una de sus consecuencias.
4. No ocurre que: $H \Rightarrow E$
Que puede leerse como: H, nuestra hipótesis abductiva, no tiene, por sí sola, a E como una de sus consecuencias.
5. $K(H)$ es consistente
Que puede leerse como: la revisión de K, resultante de agregar H a K, es consistente
6. $K(H)$ es mínima
Que puede leerse como: la revisión de $K(H)$ es mínima
7. $K(H) \Rightarrow E$
Que puede leerse como: la revisión $K(H)$ sí tiene como una de sus consecuencias a E
8. Por lo tanto: H

Antes de enumerar los 8 ítems del esquema *AKM*, Gabbay y Woods le atribuyen a éste dos ideas básicas: I1) que está estructurado para capturar la abducción consecuencialista. Esto significa que en *AKM*, la abducción está concebida como la inferencia de una hipótesis H, a partir de condiciones específicas para una noción de consecuencia, en este caso, la relación expresada por “ $\alpha \Rightarrow \beta$ ”, para cualesquier α, β . La segunda idea, I2), es que la manera estándar en que los filósofos de la ciencia interpretan la noción de consecuencia que caracterizaría a $\alpha \Rightarrow \beta$, es tomarla como una relación explicativa, expresaría que α explica β . Finalmente, en [1] se concluye que el esquema *AKM* subrepresenta la estructura lógica de la abducción:

“Proposition 12.1 (Under-representation) The *AKM*-schema under-represent the logical structure of abduction.”

Para esta conclusión los autores se apoyan en dos tesis contenidas en la siguiente aseveración del párrafo 12.1:

We are suggesting, both that the AKM-parameters are too few so that their interpretation is too narrow to afford a suitably comprehensive representation of the structure of abduction.

La distinción de las dos tesis a las que he aludido, puede quedar como sigue:

- a) El esquema AKM incluye pocos parámetros para permitir una representación adecuadamente comprensiva de la inferencia abductiva.*
- b) La interpretación que el esquema AKM supone para estos parámetros es demasiado estrecha para permitir una representación adecuadamente comprensiva de la inferencia abductiva.*

La primera de las tesis, a), se sostiene en [1] sobre la idea de que valdría la pena elucidar formalmente los vínculos explicativos que supone la relación de consecuencia en el esquema *AKM*. Si no se hacen explícitos tales vínculos o condiciones la idea de consecuencia representada por la flecha “ \Rightarrow ” no funcionaría como una elucidación de la abducción. Así, se demostraría la escasez de parámetros para la elucidación de la inferencia abductiva. La segunda tesis, b), se apoya de modo central en el supuesto de que entre los filósofos de la ciencia la interpretación estándar que elucidaría la relación de consecuencia es la interpretación explicativa de esta relación. No obstante, también se sostiene en otro supuesto: la asunción de que esta relación explicativa es caracterizable identificando las condiciones de la relación en cuestión, con las condiciones del modelo Nomológico Deductivo de Hempel y Oppenheim, de 1948, [7] (en adelante “modelo D-N”). Con estos antecedentes, el argumento para b) iría como sigue: Si ésta es la caracterización de la noción de consecuencia, es claro que *K* tendría que incluir un conjunto de leyes y *H* tendría que ser una condición inicial. No obstante, el carácter hipotético de *H* haría imposible que se tratara de una condición inicial. De lo inmediatamente dicho se sigue que la interpretación de los parámetros es demasiado estrecha para una representación adecuada.

3. Problemas en la crítica de Gabbay y Woods, en el caso de la interpretación explicativa

Defenderé en este apartado que al punto de crítica descrito en el apartado anterior le ocurre que: T1) no toma en cuenta seriamente el contexto de la discusión en el ámbito de la Filosofía de la Ciencia en sus términos más generales y, T2), no considera las diversas posiciones que sobre la teoría de la explicación se han defendido en esta disciplina filosófica.

Es casi inevitable comparar la perspectiva de Woods y Gabbay, respecto al problema de la representación adecuada de la inferencia abductiva, con el problema de la

representación formal adecuada de la explicación científica, tal como se le ha asumido a éste desde finales de la década de los 40's y hasta la fecha. Uno podría defender este punto de vista de un modo relativamente fácil como sigue. i) Las tesis a) y b) suponen, como explícitamente se enuncia en el artículo [1], que una evaluación de las capacidades del modelo pasa por dos criterios, resumidos en dos conceptos: parámetros de representación suficientes e interpretaciones adecuadas de éstos. ii) De un modo análogo, la discusión sobre la posibilidad de construir un modelo formal de explicación, a lo largo de la segunda mitad del siglo XX, corrió sobre estos mismos supuestos. Obsérvese por ejemplo, la crítica de Scriven en 1962 [8] contra la necesidad y contra la suficiencia del modelo Nomológico Deductivo propuesto por [7]. También puede consultarse la crítica de Achinstein en 1983 [9] que denunció la no inclusión formal de la noción de causalidad en los modelos de explicación; o la de van Fraassen en 1980 [10], sobre la no inclusión del contexto en los modelos formales de explicación conocidos hasta entonces. Así iii), *prima facie*, es un procedimiento adecuado el de mostrar aspectos del objeto modelado que no están incluidos en un modelo, con la intención de mostrar las debilidades de este último en su objetivo de representación.

No obstante, si se le examina con detenimiento, se puede mostrar que el procedimiento seguido en [Gabbay & Woods, 2006] puede llevar a errores. La falla básica puede describirse como la relativización del conjunto de aspectos a los que el evaluador en turno tendría permitido apelar para construir su crítica contra algún modelo. Formulado en una pregunta, el problema es ¿Cualquier aspecto no capturado por el modelo puede articular una crítica de este tipo que sea sensata? La respuesta es no. El argumento central para una respuesta negativa a esta pregunta es que el conjunto de aspectos de un objeto, sin importar si es conceptual o físico, es indeterminado. Esta es una conclusión que puede derivarse de un argumento que aparece, en otro contexto, en un texto de Putnam de 1981[11] y en uno de la misma autoría, de 1994[12]. El argumento versa sobre la indeterminación del número de objetos en que uno podría dividir cualquier ontología propuesta. El argumento es una aplicación que hace Putnam al mundo empírico, de un teorema en Teoría de Modelos: el resultado conocido como el teorema de Skolem-Lowenheim. La idea de Putnam es que, del mismo modo en que la totalidad de las verdades acerca de los objetos matemáticos que pueden expresarse en el lenguaje de las matemáticas no puede determinar a qué objetos nos estamos refiriendo, no podemos tampoco determinar la totalidad de los objetos empíricos. Se puede ver que tampoco podemos hacerlo en un dominio conceptual diferente al de las matemáticas. Así, hay una indeterminación esencial acerca de los objetos que uno puede considerar en cada realidad supuesta. Esto vale para los componentes que uno considere importantes en cada objeto modelado. Así, siguiendo esta línea de pensamiento, siempre tendremos la posibilidad de exigir nuevos aspectos que modelar. Consecuentemente, la tesis a), por sí sola, no puede tomarse como razón suficiente para que a partir de la incapacidad, que tiene uno cualquiera de los modelos considerados, de capturar alguno de los aspectos estipulados, uno pueda intentar restringir (o hacer racional) una elección entre los diversos modelos. Por lo tanto, dado el análisis presentado hasta ahora, tal como se ha expresado la tesis a) en [1], ésta es una tesis que no tiene el alcance que se le atribuye ahí mismo.

Analizaré ahora no la tesis b), sino cómo se le defiende en [1]. Puede concederse que la interpretación estándar de la abducción consecuencialista, entre los filósofos de

la ciencia, sea una relación explicativa. No obstante, lo anterior no hace que la afirmación de existencia de esa interpretación estándar deje de ser una aseveración que tendría que justificarse, lo cual no se realiza en el artículo. Sin soslayar lo anterior, lo que es claro es que esa aseveración, aun concediéndola, no implica que la caracterización de esa interpretación tenga que estar basada en el modelo D-N de explicación. Este modelo puede considerarse un modelo clásico, quizá paradigmático para los fines del debate de la segunda mitad del siglo XX, pero en ningún sentido es necesario que se le tome como *la* caracterización de la relación de explicación. Tampoco puede sostenerse que sea la caracterización estándar de la interpretación estándar entre los filósofos de la ciencia. Por el contrario, el debate en la Filosofía de la Ciencia, alrededor del problema de si es posible un modelo formal de explicación científica, ha producido una cantidad muy grande de muy diversas alternativas de modelación. Aún más, muchas de estas alternativas fueron justificadas parcialmente mediante una dura crítica contra el modelo D-N y, también, contra el modelo Inductivo-Estadístico (en adelante, “modelo I-S”) propuesto algunos años después que el D-N, en 1965 [13]. Algunos casos que pueden ilustrar lo anterior son los siguientes: la crítica de Railton en 1981 [14] contra el modelo I-S, sobre la cual se erige un modelo basado en una descripción deductiva y un *addendum* descriptivo; también la crítica de Salmon, de 1984, [15] dirigida contra la estructura argumentativa de las explicaciones, a partir de lo cual desarrolla un modelo que puede llamarse “de relevancia estadística”; otro caso muy significativo es la crítica contra la posibilidad de determinar un contexto de relevancia uniforme, que esgrime van Fraassen, 1980, [10], y el modelo de explicación que se propone ahí mismo, basado en lógica erotética. En consecuencia, no debería asumirse que la única caracterización de una interpretación explicacionista de la abducción consecuencialista sea el modelo D-N Hempeliano, ni siquiera sería conveniente asumir que es su caracterización estándar.

Todas estas consideraciones sobre la defensa de la tesis b) tienen por objetivo mostrar que, por lo tanto, no deberíamos asumir que la caracterización de la interpretación de la abducción consecuencialista deba ser el modelo D-N de Hempel y Oppenheim, ni siquiera debería considerarse a este modelo clásico la caracterización estándar de la interpretación estándar de la abducción consecuencialista. En este sentido, la defensa de la tesis b) tal como se presenta en [1], es inadecuada en el sentido de que intenta rechazar la interpretación explicacionista con base en una caracterización no representativa de la relación explicativa.

Ahora bien, Gabbay y Woods pueden seguir manteniendo la tesis b), si hallaran una forma de sostener que las interpretaciones de los parámetros en *AKM* son demasiado estrechas para permitir una representación adecuadamente comprensiva de la estructura de la inferencia abductiva. Si, efectivamente, *AKM* asumiera el modelo D-N Hempeliano como la caracterización de su interpretación explicacionista de la abducción consecuencialista, entonces *AKM* estaría en problemas. Notemos, de cualquier forma, que hemos mostrado que ello no necesariamente pondría en problemas a toda interpretación explicacionista de la versión consecuencialista de la abducción.

Un examen de este problema más particular contra *AKM*, requeriría un análisis más detallado de las propuestas que conformarían, según Gabbay y Woods, el esquema *AKM*; es decir, se necesitaría un examen de las propuestas de Aliseda, Kuipers, Kowalski, Kakas y Magnano. No realizaré este análisis en este artículo. Sin embargo, me interesa enfatizar que en términos generales, no sólo en *AKM*, sino en varias otras propuestas en Inteligencia Artificial que intentan caracterizar la explicación, existe una

tendencia a la sobresimplificación de la noción de explicación, identificando los ítems importantes de esta noción con las condiciones de los modelos D-N e I-S, los que podríamos llamar “modelos clásicos de la explicación”. Lo que es más importante, en algunas propuestas, no solo se asumen los modelos clásicos de Hempel y Oppenheim como la interpretación estándar de la explicación en Filosofía de la Ciencia, sino que se omite gran parte de los problemas planteados en el debate que ha dado esta disciplina filosófica. En el caso de Aliseda, [2], que puede entenderse quizá como la propuesta más importante y dominante dentro del enfoque formal tradicional de la abducción, se dedica un análisis centrado en estos modelos clásicos y una consideración de algunos problemas planteados por Salmon. Aunque la propuesta contenida en [2] debe considerarse una teoría lúcida y fundacional en el estudio de la abducción, en lo que respecta a su vinculación con la explicación se mantiene en los límites todavía estrechos de la caracterización de los modelos clásicos.

El caso del enfoque *GW*, el propuesto por Gabbay y Woods, no corre mejor suerte en relación con la noción de explicación. Una muestra de esto es, tal como lo he descrito aquí, la manera en que se organiza en [1] la defensa de la tesis b); pero también puede verse esta tendencia en algunas asunciones que pueden tomarse de un artículo de Woods de 2007 [16]. La propuesta del enfoque *GW* para la abducción tiende no solo a asumir la reducción de la caracterización de la explicación a los modelos clásicos, sino a minimizar el papel de la explicación en la abducción.

Los problemas para sostener a) y b), hacen que la conclusión de que el esquema *AKM* sub-representa la estructura lógica de la abducción, la proposición 12.1, aparezca como no bien fundada, por lo menos por esta vía.

El examen presentado hasta aquí de este punto de crítica de [1], debe entenderse como una defensa de T1), a saber, que la crítica en relación con la explicación no toma en cuenta seriamente el contexto de la discusión en el ámbito de la Filosofía de la Ciencia en sus términos más generales. También es una defensa de T2), es decir, que no considera las diversas posiciones que sobre la teoría de la explicación se han defendido en esta disciplina filosófica.

4. Las intuiciones conceptuales: Inteligencia Artificial y Filosofía de la Ciencia

En el caso de la tesis b), la asunción del modelo D-N es crucial para el fallo de la argumentación en ese apartado del artículo de Woods y Gabbay y se debe, en gran medida, a la sobresimplificación que suele hacerse en otras disciplinas, en este caso en la Inteligencia Artificial, de las discusiones epistemológicas acontecidas en el ámbito de la Filosofía de la Ciencia. Por el contrario, quizá deberíamos confiar en que una actitud más abierta y participativa entre las dos disciplinas, Inteligencia Artificial y Filosofía de Ciencia, podría venir bien al enriquecimiento de las producciones teóricas de ambas. En el caso que nos ocupa ahora, y en especial con relación al caso de la tesis b), bastaría con diversificar nuestra idea de los problemas involucrados en el debate que sobre la explicación se ha generado en la Filosofía de la Ciencia, y conocer los diversos modelos de explicación que se han propuesto en él. En cambio, para el caso de la tesis a), es necesario notar que, siguiendo a Putnam, el que uno pueda usar cualquier aspecto del objeto modelado para realizar una crítica contra un modelo dado, no es indicativo

de la imposibilidad de discutir sobre la adecuación de un modelo para ciertos fines de representación. Sabemos que la caracterización de un conjunto de aspectos sobre los que se pudiera establecer un criterio de elección o crítica entre modelos de la abducción es un asunto bien problemático, pero esto no quiere decir que no puedan establecerse, con algún criterio razonable, algunos aspectos a los que uno podría apelar. La clave está en que esos aspectos, como ocurre en el caso de la discusión sobre la explicación científica en Filosofía de la Ciencia, deberían tener algún grado de atrincheramiento en la cultura general que subyace a esa discusión. La discusión sobre la explicación, en Filosofía de la Ciencia, se dio en el marco de una serie de supuestos alrededor de la explicación, que estaban más o menos atrincherados en la cultura científica y filosófica pertinente al caso. Puede confrontarse para este asunto el texto iluminador Salmon [17]. Por ejemplo, la idea de la necesidad de una vinculación causal en la explicación o la idea de la necesidad de la apelación a leyes, no fueron una construcción ad-hoc de un ítem particular de ese debate en Filosofía de la Ciencia. Por el contrario, fueron aspectos recogidos y precisados a partir de las intuiciones de quienes, a lo largo de siglos, habían tocado el tema de la explicación. En la propuesta defendida en [1] la reconstrucción histórica de estos aspectos de la abducción que deberían modelarse, no parece bien desarrollada. Se hacen algunas pocas menciones a la tradición de Peirce en la abducción, sin un análisis de esa tradición. En todo caso, incluso, la propuesta peirciana parece más coincidente con la posición del esquema *AKM*, que con el modelo que ellos defienden, *GW*. Esto es cierto por lo menos en lo que respecta a la estructura con la que en ocasiones Peirce describió la abducción, pero también respecto a la tendencia de este pensador a vincularla con las relaciones de causalidad.

5. La relevancia explicativa: Inteligencia Artificial y Filosofía de la Ciencia

Finalmente, daré apenas un bosquejo de una aportación positiva para toda esta discusión. Un avance en la dirección correcta para la construcción de un contexto que sirviera de contraste para la elección de aspectos a tomar en cuenta en la elección racional entre modelos de la abducción podría partir de las conexiones de la abducción con el programa general de la disciplina de la Inteligencia Artificial. Un análisis importante en esta agenda de reconstrucción de la tradición, en particular sobre las nociones de relevancia en Inteligencia Artificial, podría iniciar prometedoramente con un análisis de los problemas del *frame* y de la cualificación, los cuales están en la base de la tradición en Lógica No Monotónica para lograr cierta clase de representaciones en Inteligencia Artificial. Particularmente, un examen de las discusiones de McCarthy y Hayes en 1969 [18] y del artículo posterior, 1977, de este primer autor [19] seguramente sería un estudio clave. Vale decir también que Minsky, en 1974 [20] también hizo algunas reflexiones que deberían reconsiderarse y que vinculan, con toda claridad, la Inteligencia Artificial con la Filosofía de la Ciencia. Exploremos el caso del problema del *frame* y el de la cualificación con mayor detalle, tomando en cuenta su caracterización en [18] y [19].

El problema del *frame*, puede pensarse como el problema de saber cuáles supuestos deben modificarse y cuáles no, con base en nueva información. En el mismo tenor, el problema de la cualificación puede pensarse como el problema de saber cuáles son los

supuestos posibles que deben tomarse en cuenta ante la resolución de un problema. Visto desde un punto de vista un poco más abstracto, ambos problemas tratan de la forma en que determinamos, a partir de una situación dada (sea una situación inicial o el resultado del cambio de esta situación inicial), lo que es asumible. Una forma de interpretar estos problemas, así unidos es mediante la siguiente pregunta: ¿Cómo determinar lo que es pertinente para sacar consecuencias a partir de una situación dada? Pueden así plantearse dos subcasos de este problema: Cuando la situación se toma como punto de partida; es decir, sin considerar aumentos de información (cualificación) y, el caso contrario, cuando debe tomarse en cuenta un aumento de información en el conjunto-base de datos (*frame*). Podemos llamar a este problema compuesto “el problema general de la relevancia (o de la pertinencia)”.

Un detalle, sin embargo, es imprescindible de tomarse en cuenta: el problema de la pertinencia general no puede caracterizarse cabalmente si no se supone que la determinación que exige tiene que hacerse con base en información incompleta. El problema de la relevancia general surge en el marco de una situación con información incompleta. Una situación así, si se aborda de modo consecuencialista, implica dos subproblemas más:

- 1) El problema de determinar estructuras lógicas que permitan saltar a la conclusión o, en otras palabras, que permitan conjeturar, en un razonamiento.
- 2) El problema de manipular razonamientos basados en supuestos falibles.

Lo que está de fondo en ambos subproblemas es cómo llegar a conclusiones de manera “sensata”, a partir de premisas que son insuficientes. Si entendemos “conjetura” como un razonamiento que supone un salto a la conclusión, la cuestión central se traduce en cómo hacer y manipular conjeturas sensatas. Podemos llamar a este problema general, el “problema de la conjetura”. El problema de la conjetura puede caracterizarse como una de las formas de abordar, desde un punto de vista lógico, el problema general de la relevancia. Así, aparece con mayor naturalidad el desiderata de que los problemas surgidos en Inteligencia Artificial sean considerados como problemas que involucran discusiones de Lógica pero también de Epistemología, pues el problema general de la relevancia tiene profundas raíces en cuestiones de índole epistemológica.

El problema de la conjetura, una forma de abordar el problema general de la relevancia, está estrechamente vinculado tanto con el problema de la modelación de la abducción, como con el problema de la modelación de la explicación. Las condiciones que se añadan a los núcleos esquemáticos de los modelos de una y otra noción deberían capturar adecuadamente los rasgos de relevancia atinentes a cada caso. En el primer caso, el de la abducción, para concluir la hipótesis, en el segundo, la explicación, para concluir el hecho que se pretende explicar. Ambos problemas de modelación requerirán de algún tipo de solución al problema general de la relevancia y, si se les aborda de modo consecuencialista, esto se traducirá en la búsqueda de una solución al problema de la conjetura. Dos insumos para abordar estos problemas son pertinentes. Por un lado, reflexiones sobre las implicaciones que tienen ciertos formalismos no monotónicos en la noción de racionalidad (las que considera reflexiones en el límite de la Filosofía de la Lógica y la Epistemología) y, por otro lado, reflexiones epistemológicas sobre las condiciones que puedan constituir una noción de relevancia vinculada a una conjetura particular, por ejemplo, la conjetura de un hecho particular a partir de una teoría. Un ejemplo del primer tipo de reflexiones es el conjunto de disquisiciones elaboradas por

Morado en [21]. Ejemplos del segundo tipo de reflexiones pueden encontrarse en las propuestas de Salmon en [15] y de van Fraassen en [10].

6. Conclusiones

El bosquejo de la sección anterior muestra, en una dirección, cómo el problema general de la relevancia y la forma en que se aborda formalmente desde la Inteligencia Artificial, puede tener importantes consecuencias en la modelación de la explicación, desde el ámbito de la Filosofía de la Ciencia. En otra dirección, ilustra la pertinencia de tomar en cuenta los amplios estudios que sobre la noción de relevancia explicativa se han hecho desde la Filosofía de la Ciencia, a la hora de la construcción de restricciones de relevancia en un modelo de abducción que se considere, desde el punto de vista de la Inteligencia Artificial, vinculado a la noción de explicación.

A través de las reflexiones contenidas en este artículo, se pueden establecer tentativamente las siguientes aseveraciones:

- 1) El punto de crítica de Gabbay y Woods vinculado a la caracterización explicacionista de la interpretación consecuencialista de la explicación, tiene fallas que deben considerarse.
- 2) Estas fallas muestran una deficiencia en la interacción entre investigaciones de dos diferentes disciplinas: Inteligencia Artificial y Filosofía de la Ciencia.
- 3) Un caso posible de cooperación, a nivel metodológico, entre ambas disciplinas, es el uso de conceptos intuitivos atrincherados en torno a la noción de explicación y de abducción.
- 4) Un caso plausible de revisión es el de las condiciones de relevancia para una interpretación consecuencialista de la abducción. Por un lado, es importante recuperar las intuiciones básicas del problema general de la relevancia en Inteligencia Artificial, por otro, reflexiones epistemológicas sobre los formalismos y, finalmente, reflexiones sobre las condiciones de relevancia de una explicación.
- 5) Puede resultar conveniente unir los esfuerzos de la Inteligencia Artificial y de la Filosofía de la Ciencia, por lo menos en el problema de la modelación de la abducción y en el de la modelación de la explicación.

Agradecimientos. Agradezco el apoyo para esta investigación, por parte de la Secretaría de Ciencia y Tecnología del Distrito Federal a través del proyecto PI2014-47.

Referencias

1. Gabbay, Dov, Woods, John: *Advices on Abductive Logic*. *Logic Journal of IGPL*. 14(2) pp. 189–219 (2006)
2. Aliseda Llera, Atocha: *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence*. Amsterdam: Institute for Logic, Language and Computation (1997)
3. Kowalski, R. A.: *Logic for Problem Solving*. New York, Elsevier (1979)

4. Kuipers, Theo, A.F.: Abduction aiming at empirical progress of even truth approximation leading to a challenge for computational modelling. *Foundations of Science* (1999)
5. Magnani, Lorenzo, *Abduction, Reason and Science: Processes of Discovery and Explanation*. New York: Kluwer, Plenum (2001)
6. Kakas, A., Kowalski, R.A., Mancarella, P.: Knowledge assimilation and abduction. In: *Proceedings of the European Conference on Artificial Intelligence, ECAI, Springer-Verlag* (1990)
7. Hempel, Carl G., Oppenheim, Paul: *Studies in the Logic of Explanation*. *Philosophy of Science*, vol. 15, pp. 567–579 (1948)
8. Scriven, Michael: *Explanations, Predictions, and Laws*. University of Minnesota press, Minneapolis (1962)
9. Achinstein, Peter: *The Nature of Explanation*. Oxford University Press, New York (1983)
10. Van Fraassen, Bas: *The Scientific Image*, Oxford University Press, New York (1980)
11. Putnam, Hilary: *Reason, Truth and History*. Cambridge University Press (1981)
12. Putnam, Hilary: *The Dewey Lectures 1994: Senses, Non senses and The Senses: An Inquiry into the Powers of the Human Mind*. *The Journal of Philosophy Inc.*, New York (1994)
13. Hempel, Carl: *Aspects of Scientific Explanation*. The Free Press, New York (1965)
14. Railton, Peter: A Deductive-Nomological Model of Probabilistic Explanation. *Philosophy of Science*, 45 (1978)
15. Salmon, Wesley: *Scientific Explanation and the Causal Structure of the World*. Princeton University Press (1984)
16. Woods, John: Ignorance and Semantic Tableaux: Aliseda on Abduction. *Theoria*. vol. 60, pp. 305–318 (2007)
17. Salmon, Wesley: *Four decades of Scientific Explanation*. University of Minnesota (1989)
18. McCarthy, John, Hayes, P.: Some Philosophical Problems from the Standpoint of Artificial Intelligence. B. Meltzer and D. Michie (eds.), *It Machine Intelligence*, Vol. 4, Edinburgh, N.Y., p. 468 (1969)
19. McCarthy, John: Epistemological Problems of Artificial Intelligence. In: *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1038–1044 (1977)
20. Minsky, Marvin: A frame for Representing Knowledge. AI memo, 306 (1974)
21. Morado, R.: Problemas filosóficos de la lógica no monotónica. Raúl Orayen & Alberto Moretti (eds.) *Filosofía de la lógica*, Enciclopedia Iberoamericana de Filosofía, Vol. 27, Edit. Trotta, Madrid (2004)

Minería de opinión en blogs financieros para la predicción de tendencias en mercados bursátiles

Sergio Hernández, Sabino Miranda-Jiménez, Elio Villaseñor,
Eric Sadit Tellez, Mario Graff

INFOTEC - Centro de Investigación e Innovación en Tecnologías de la Información y
Comunicación, Aguascalientes, México

kertr3@hotmail.com, {sabino.miranda, elio.villasenor,
eric.tellez,mario.graff}@infotec.mx

Resumen. El análisis de redes sociales para el estudio de mercados financieros se ha vuelto un tema de investigación y desarrollo de herramientas que permite a los agentes financieros usar las opiniones de la gente para aumentar la precisión en las predicciones de mercado. Nuestra investigación se enfoca en la predicción de la tendencia de índices financieros usando la minería de opinión, basado en el análisis de blogs especializados en finanzas para el idioma inglés. Los comentarios vertidos en estos blogs son clasificados en términos de su opinión respecto a la tendencia de mercado (a la alza, estable o a la baja). Se evalúan distintas técnicas de aprendizaje computacional y minería de textos para la clasificación de los comentarios realizados durante un periodo de tres meses. Los resultados obtenidos muestran que este análisis puede ser incorporado como un factor en la toma de decisión de los agentes financieros y mejorar la precisión de sus proyecciones.

Palabras clave: minería de opinión, *algorithmic trading*, aprendizaje computacional.

1. Introducción

En los últimos años, la minería de opinión se ha usado en diferentes campos de aplicación como reseñas de películas, ranking de productos, prestigio de personalidades (políticos, figuras del medio de espectáculo, etc.), entre otros [1, 2]. En particular, hay un fuerte interés de aplicar los métodos de minería de opinión en el ámbito económico-financiero [3,5] para crear o mejorar los métodos de predicción con base en la información de blogs especializados en el tema, donde participan activamente agentes (*traders*, especuladores, *hedgers*) o líderes de opinión (analistas financieros). Por otro lado, dado que se han creado nuevas formas de realizar transacciones financieras mediante el uso de computadoras con más capacidades y mejor desempeño, surgió el área de aplicación llamada *Algorithmic Trading* (AT), también conocida como *trading* automático.

El AT es el uso de plataformas electrónicas para realizar transacciones financieras con algoritmos en variables que incluyen tiempo, precio y volumen de los índices financieros [4]. Los dos enfoques principales del AT son análisis de alta frecuencia y el análisis cuantitativo. Uno de los intereses es el análisis cuantitativo o también conocido como *Quantitative Trading* (QT). El QT es generalmente usado por instituciones financieras, donde se manejan grandes volúmenes de transacciones que involucran la compra y venta de cientos de miles de instrumentos financieros (acciones, futuros, etc.). El QT utiliza datos históricos que se remontan en el tiempo para crear una proyección a futuro, analiza la estructura, encuentra patrones y tendencias de mercado.

Recientemente, los métodos de QT están incorporando información de medios sociales (Twitter, Facebook, blogs, etc.) para mejorar o crear estrategias financieras y predicciones del mercado bursátil [6]. En el dominio de las finanzas, muchos *bloggers* publican sus opiniones sobre compañías específicas y sobre los mercados financieros en general.

Los medios sociales especializados, en particular los blogs dedicados al tema financiero, tienen la ventaja que sus usuarios están relacionados con este ámbito y es probable que sus opiniones y su conocimiento del mercado bursátil este expresado en dichos mensajes. Por lo que se espera que este conocimiento pueda ser útil para mejorar las predicciones de los algoritmos sobre el comportamiento de los instrumentos financieros.

En los diferentes blogs especializados, se discuten diversos temas como: el del mercado financiero, de las acciones, estrategias de operación, divisas, entre muchos otros; dicha información puede usarse para realizar predicciones del mercado bursátil basadas en el análisis del sentimiento de las opiniones [8, 9]. Por ejemplo, se han realizado estudios sobre la red social Twitter que demostró una conexión entre las relaciones de trabajo de las personas y el sentimiento creado por medio de la red de Twitter referente a las acciones del mercado, donde el sentimiento de la red a su vez influye en el movimiento de ciertas acciones [10].

También, se han realizado varios estudios acerca de la relación entre la búsqueda en línea y el comportamiento de acciones en el mercado [11, 12]. En [13] se encontró que el incremento en la intensidad de la búsqueda de acciones en Japón, incrementa la actividad de operaciones realizadas en dichas acciones. Otro tipo de estudio [14] relacionado con la minería de opinión, se basa en los títulos de noticias financieras y no en el texto completo del documento para la predicción del mercado de divisas.

Otro estudio en Twitter [6] demostró una relación entre el sentimiento y la ganancia en las acciones del mercado, así como el volumen de mensajes con el volumen de transacciones bursátiles realizadas. En [7] se muestra un método de AT que considera el análisis de sentimientos de comentarios en un blog que produce mejores ganancias que los métodos tradicionales de compra-venta, ya que obtuvieron un aumento de 56.28% en la precisión de la tendencia.

Nuestro trabajo se enfoca en la minería de opinión de los comentarios subjetivos vertidos en blogs especializados del ámbito financiero (*Business Insider*, *Econbrowser*, *Dealbreaker*, *Investing*, etc.). Esto involucra la clasificación de dichos comentarios en las siguientes clases: a la alza, estable, a la baja. Además nuestro enfoque incorpora un

lexicón de palabras clasificadas de acuerdo a las clases de tendencia mencionadas. Este lexicón se construyó manualmente.

El resto del artículo está organizado como sigue. En la sección 2, se describe, la metodología de nuestro enfoque. En la sección 3, se presentan los resultados obtenidos. Por último, en la sección 4, se presentan las conclusiones y el trabajo futuro.

2. Metodología

El enfoque que se propone es incorporar información de medios sociales (blogs especializados) a los métodos de QT para mejorar sus predicciones. Para lograr esto, se plantea el esquema de la Fig. 1, el cual es un enfoque supervisado tradicional, donde se cuenta con información etiquetada manualmente (corpus de entrenamiento) y se usa como entrada en la etapa de entrenamiento, donde se construye el modelo de predicción. En este artículo, consideramos solamente la información de blogs especializados y, por ahora, no incorporamos la información de los índices financieros dentro del modelo de predicción.

2.1. Esquema general

En esta parte del trabajo, se considera únicamente la clasificación de los comentarios que indican si los índices financieros están a la alza, a la baja o se mantienen estables.

Se presenta la construcción del corpus financiero etiquetado por un experto, así como el preprocesamiento de la información, que consiste en la limpieza de datos de los comentarios, y la clasificación de estos por medio de los métodos supervisados propuestos; además, se incorpora conocimiento adicional al modelo de predicción a través de un diccionario de palabras etiquetado manualmente que representan tendencias del mercado financiero.

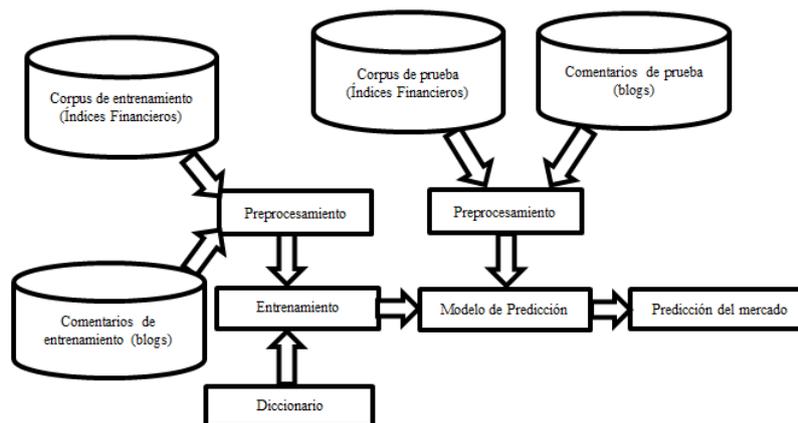


Fig. 1. Esquema general de la solución propuesta.

2.2. Índices financieros

Un índice financiero es un número abstracto que representa el movimiento en conjunto de varios activos financieros que lo componen (acciones, bonos, monedas, *commodities*, etc.). Cada uno de estos activos tiene un peso relativo dentro del índice, medido según parámetros previamente establecidos a la creación del índice. Entonces, ante un movimiento de precios del activo se moverá el índice y la variación del índice será mayor o menor según el peso del activo. Los Índices más conocidos son los de las bolsas de valores como el NYSE, Nasdaq, Dow Jones, SP&500, Merval, Nikkei, etc.

2.3. Creación del corpus

La información que se utilizó para la creación del corpus proviene del blog financiero *investing.com* [15] sobre el índice S&P500 [16]. El blog de discusión *Investing.com* es un portal de internet que proporciona noticias, análisis, datos técnicos sobre los mercados financieros globales donde los usuarios pueden obtener una fuente óptima integral de características innovadoras en diferentes instrumentos financieros.

Para la generación del corpus de entrenamiento de comentarios financieros, se automatizó la descarga de comentarios del blog para el activo ‘futuros’ referente al índice financiero S&P500. El etiquetado fue realizado por un experto en el tema financiero. Se tomaron 3 meses de comentarios (06-ene-2015 a 03-mar-2015). Se etiquetó cada comentario con alguna de las siguientes clases: a la alza, estable o a la baja; debido a que el precio de cierre del índice S&P500 solo puede tomar alguna de las siguientes posiciones con respecto a su estado anterior: mayor al anterior (alza = 1), Igual al anterior (estable = 0) y menor al anterior (baja = -1). Por ejemplo, en la Tabla 1, el comentario 3 clasificado como 1, contiene palabras como *bull*, que hacen referencia generalmente a una tendencia a la alza del mercado. En el comentario 2, la palabra *short* denota normalmente una tendencia a la baja y se clasifica como -1. En el comentario 4, no hay alguna expresión o palabra significativa que marque una tendencia, por lo cual se clasifica como 0.

Dentro del etiquetado el experto tuvo problemas al clasificar ciertos comentarios debido a la falta de contexto o falta de conocimiento sobre la posición en que se encontraba el mercado al momento de publicarse el comentario; por ejemplo, en el siguiente comentario “*Looks like the 2100 support will be broken today -- get ready*”; no se tiene suficiente contexto para decidir si el mercado tiene una tendencia a la alza o a la baja, por lo que se decidió clasificar este comentario de forma estable (0).

Tabla 1. Ejemplos de comentarios etiquetados.

Num.	Comentario	Clase
1	if that's what you believe you need to read this : http://finance.yahoo.com/news/us-running-room-store-oil-171025359.html	0
2	low risk entry for a scalp short	-1
3	end of this bull run is near..	1
4	Sweet!	0
5	Shorting again.. sorry couldn't resist. Lol	-1

El corpus recopilado consta de un total 837 comentarios etiquetados; y contiene 1,884 palabras únicas.

En la Tabla 1 se muestran algunos ejemplos etiquetados por el experto del dominio.

2.4. Descripción del diccionario

Nuestro enfoque incorpora el conocimiento de un experto al modelo por medio de un diccionario de palabras clave relacionadas con la alza o la baja en la tendencia del índice financiero. Las palabras clave fueron etiquetadas por el experto con las clases a la alza (1) y a la baja (-1). El diccionario cuenta con 52 palabras/frases únicas usadas comúnmente en los foros especializados en temas financieros. En la Tabla 2, se presentan algunos ejemplos de las palabras del diccionario.

El etiquetado fue añadido al conjunto de entrenamiento para comparar si una metodología sin diccionario o con diccionario obtenía una mejor predicción.

Tabla 2. Ejemplo de palabras del diccionario propuesto

Palabra	Clase	Palabra	Clase	Palabra	Clase	Palabra	Clase
<i>Bear trap</i>	1	<i>Go north</i>	1	<i>Heading below</i>	-1	<i>Niagara</i>	-1
<i>Bull trap</i>	-1	<i>Going up</i>	1	<i>overbought</i>	-1	<i>Bearish</i>	-1
<i>Oversold</i>	1	<i>Climbing up</i>	1	<i>Bottom line</i>	-1	<i>Drown</i>	-1
<i>Downtrend</i>	-1	<i>Trend head</i>	1	<i>Going down</i>	-1	<i>Bottom</i>	-1
<i>Heading above</i>	1	<i>bullish</i>	1	<i>downtrend</i>	-1	<i>Rush sell</i>	-1
<i>Upper line</i>	1	<i>breakup</i>	1	<i>Bull trap</i>	-1	<i>Break down</i>	-1
<i>Bear trap</i>	1	<i>Up leg</i>	1	<i>Run down</i>	-1	<i>Head down</i>	-1

2.5. Métodos de clasificación

A continuación se presentan las herramientas y métodos para para la solución del problema planteado, los diferentes pasos que se siguieron para procesar el corpus construido; además, se presentan los clasificadores utilizados para la clasificación de los comentarios.

Preprocesamiento de los datos. En esta etapa, se realiza una reducción del vocabulario usado por medio del filtrado de las palabras de contenido. Este proceso se aplica a todos los comentarios que se consideran para el proceso de clasificación. Para reducir el vocabulario, usamos un filtrado basado en lematización, y solo usamos las palabras de contenido, es decir, sustantivos, adjetivos, verbos y adverbios; también usamos las interjecciones debido a que contienen un contexto emotivo; también se eliminan las *stop words* (preposiciones, determinantes, etc.).

Para realizar esta etapa, usamos el etiquetador de partes de oración de Stanford para el idioma inglés [17] y usamos solo los lemas de las palabras. Para el filtrado de las palabras usamos las etiquetas definidas por el *Penn Tree Bank Project* para el idioma

inglés [18]. Por ejemplo, en el comentario “*Looks like the bulls returned after lunch*” que indica una tendencia a la alza, es etiquetado como *look_VBZ like_IN the_DT bull_NNS return_VBD after_IN lunch_NN*.

Para realizar el filtrado de palabras, se usa el inicio de las etiquetas de cada palabra, solo consideramos las palabras de contenido como se mencionó. Para el ejemplo, se seleccionaron del comentario los sustantivos (*_NN*), y verbos (*_V*); se descartaron las preposiciones (*_IN*) y los determinantes (*_DT*), no se considera como parte del mensaje final. De esta forma, el contenido del mensaje resultante sería “*look bull return lunch*”. Estas palabras son las que se consideran como características (*features*) para la etapa de entrenamiento de los modelos.

Modelo de predicción. En nuestros experimentos usamos dos clasificadores, a saber, *Support Vector Machine* (SVM) [19] y Naive Bayes (NB) [20] que usualmente se aplican en minería de opinión [1, 2]. Se utilizó la librería del software *Python* llamada *sklearn* [21] que implementa ambos clasificadores.

Para el clasificador SVM, se aplicaron 3 diferentes tipos de kernel: el primero, una función de base radial dada por $\exp(-\gamma|x - x'|^2)$ con parámetro gamma 1; en el segundo caso, se aplicó un kernel polinomial de grado $d=3$ expresado con el polinomio de la forma $(\gamma\langle x, x' \rangle + r)^d$ con un coeficiente $r=0$ y gamma 1; para el último caso, se utilizó un kernel lineal $\langle x, x' \rangle$.

En el caso del clasificador probabilístico bayesiano, su ventaja es que solo se requiere de una pequeña cantidad de datos de entrenamiento para estimar los parámetros (las medias y las varianzas de las variables) para la clasificación.

Para cada uno de los clasificadores, se aplicó una evaluación de *ten-fold cross-validation* con 2 métricas diferentes, que nos dice que tan independientes son nuestros datos de prueba contra los datos de entrenamiento. Se utilizó el 90% de los datos como entrenamiento, usando una función de la misma librería *sklearn* (*ShuffleSplit*) que toma diferentes muestras del total de datos para obtener el porcentaje requerido. Una vez obtenido el conjunto de entrenamiento se evalúa con el 10% restante de los datos, los resultados fueron promediados sobre las 10 iteraciones realizadas.

3. Resultados

Los resultados que se muestran en la Tabla 3, son obtenidos después de realizar la evaluación *ten-fold cross validation*. Se usaron medidas estándares de evaluación para los clasificadores como Precisión y *Recall* así como la medida-F, que es la media armónica de Precisión y *Recall* [22]. Donde la Precisión se define como el cociente del número de comentarios correctamente clasificados a la clase *X* entre el número de comentarios que se atribuyeron a la clase *X*. *Recall* se define como el cociente del número de comentarios correctamente clasificados a la clase *X* entre el número total de comentarios de la clase *X*.

Se realizaron principalmente dos experimentos para cada uno de los clasificadores, donde se hace uso del diccionario propuesto como conocimiento adicional y sin uso del diccionario. En la Tabla 3, se presentan los porcentajes promedio de los resultados obtenidos con tres diferentes núcleos (kernel) para el clasificador SVM. Se puede observar

que tanto el kernel de función de base radial, como el de base polinomial de grado tres presentan un promedio de precisión del 38.45% sin el uso de diccionario. En el caso del kernel lineal sin diccionario la precisión sube en aproximadamente 17 puntos porcentuales, el promedio final es de 55.12%, por lo que, es el kernel con mejor precisión.

En el caso del clasificador Naive Bayes (NB) sin uso del diccionario, se obtuvo un promedio de precisión del 55.11%, el resultado es aproximado al obtenido por el clasificador SVM con kernel lineal. El mejor resultado obtenido fue tanto del SVM lineal como del NB dando un promedio de precisión final de 57.77% incorporando el diccionario.

Tabla 3. Resultados de la validación.

Clasificador	Kernel	Sin diccionario			Con diccionario		
		Precisión	Recall	Medida-F	Precisión	Recall	Medida-F
SVM	Función radial	38.45	35.48	18.58	34.77	33.96	17.20
SVM	Polinomio	38.45	35.48	18.58	32.00	33.96	17.20
SVM	Lineal	55.12	51.36	51.04	57.77	56.03	55.83
NB	–	55.11	48.74	48.18	57.77	51.70	50.36

4. Conclusiones y trabajo futuro

En este trabajo se exploró, el uso de un blog especializado en finanzas como fuente de información para realizar análisis de sentimientos en las tendencias del mercado bursátil; también se presentó la creación de un corpus con más de 800 comentarios etiquetados que representan, según su contenido, la percepción subjetiva del mercado financiero. En el experimento se consideraron los precios de cierre del instrumento futuro en el índice S&P500 para etiquetar los comentarios, puesto que es lo que más prevalece en los comentarios de dicho blog.

También se evaluó el desempeño de dos clasificadores para esta tarea: una máquina de soporte de vectores (SVM) con tres diferentes núcleos y un clasificador bayesiano. Los mejores resultados obtenidos fueron por el clasificador SVM con un kernel lineal y por el clasificador Naive Bayes. En ambos casos, se usó el diccionario de dominio específico de palabras, que denotan una tendencia en el mercado bursátil, como conocimiento adicional de los clasificadores, con lo que se observa que el diccionario ayuda a tener mejores resultados.

Como trabajo futuro, se propone probar más clasificadores para tener una comparativa más amplia al realizar la clasificación de los comentarios, así como extender el diccionario de palabras que denotan alguna tendencia en el mercado bursátil.

Referente a la etiquetación del corpus de comentarios financieros, se identificó que el experto humano tiene dificultad al etiquetar la clase de tendencia en el mercado debido a que no cuenta, en ocasiones, con el suficiente contexto entre el comentario y el

mercado bursátil. En orden para enfrentar este problema se propone, como trabajo futuro tomar datos históricos en un lapso de tiempo referente al instrumento o índice financiero que se desea analizar y presentarlos al etiquetador experto como contexto para que tome una mejor decisión al momento de clasificar el comentario. Además, se considera usar otros blogs orientados a finanzas así como considerar otros índices financieros (NASDAQ, DOW JONES, etc.).

Agradecimientos. Este trabajo fue parcialmente apoyado por el gobierno de México (Cátedras Conacyt, SNI).

Referencias

1. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, vol. 2, pp. 1–135 (2008)
2. Viveros-Jiménez, F., Sidorov, G., Castillo Velásquez, F., Castro- Sánchez, N., Miranda-Jiménez, S., Treviño, A., Gordon, J.: Sondeos automatizados en las redes sociales a través de la minería de opinión. *Komputer Sapiens*, vol. 2 (2012)
3. Handbook of Financial Data and Risk Information. Eds. Margarita S. BroseMark D. FloodDilip Krishna, Deloitte & Touche, LLPBill Nichols, Cambridge University Press (2014)
4. Lin, T. C.: The New investor. *UCLA L. Rev.*, vol. 60, p. 678 (2012)
5. Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., Gordon, J.: Empirical Study of Machine Learning Based Approach for Opinion Mining in Tweets. *LNAI 7630*, pp. 1–13 (2012)
6. T. O. Sprenger, A. Tumasjan, P. G. Sandner y I. M. Welp: Tweets and trades: The information content of stock microblogs. *European Financial Management*, vol. 20 (5), pp. 926–957 (2014)
7. Z. Chen, X. Du.: Study of stock prediction based on social network. In: International Conference on Social Computing (SocialCom), IEEE, pp. 913–916 (2013)
8. Grayson, M. R., Kwak, M., Choi, A.: Automated platform for aggregation and topical sentiment analysis of news articles, blogs, and other online publications. *Issues in Information Systems*, vol. 15, no. 2 (2014)
9. Van de Kauter, M. Breesch, D., Hoste, V.: Fine-grained analysis of explicit and implicit sentiment in financial news articles. *Expert Systems with Applications*, vol. 42, no. 11, pp. 4999–5010 (2015)
10. Zhang, W., Li, C., Ye, Y., Li, W., Ngai, W.: Dynamic business network analysis for correlated stock price movement prediction. *Intelligent Systems, IEEE*, vol. 30, no. 2, pp. 26–33 (2015)
11. X. Li, H. Xie, L. Chen, J. Wang, X. Deng.: News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, vol. 69, pp. 14–23 (2014)
12. Fung, S., Tsai, S.-C.: Stock market-driven investment: new evidence on information, financing and agency effects. *Applied Economics*, vol. 47, no. 27, pp. 2821–2843 (2015)
13. Takeda, F., Wakao, T.: Google search intensity and its relationship with returns and trading volume of japanese stocks. *Pacific-Basin Finance Journal*, vol. 27, pp. 1–18 (2014)
14. Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., Ngo, D. C. L.: Text mining of news-headlines for {FOREX} market prediction: A multi-layer dimension reduction algorithm

- with semantics and sentiment. *Expert Systems with Applications*, vol. 42, no. 1, pp. 306–324 (2015)
15. Investing.com. <http://www.investing.com/indices/us-spx-500-futures-commentary>. Consultado el 1 abril de 2015
 16. S&P500. http://es.wikipedia.org/wiki/S%26P_500. Consultado el 1 mayo de 2015
 17. De Marneffe, M. C., MacCartney, B., Manning, C. D.: Generating typed dependency parses from phrase structure parses. In: *Proceedings of LREC*, vol. 6, pp. 449–454 (2006)
 18. Santorini, B.: Part-of-speech tagging guidelines for the Penn treebank project. (1990)
 19. Cortes, C., Vapnik, V.: Support-vector networks. *Machine learning*, vol. 20, no. 3, pp. 273–297 (1995)
 20. Murphy, K. P.: Naive Bayes classifiers. University of British Columbia (2006)
 21. Python Scikit-Learn. <http://scikit-learn.org/stable>. Consultado el 15 de enero de 2015
 22. Manning, C., Schütze, H.: *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA (1999)

Ambiente inteligente de aprendizaje con manejo afectivo para Java

Ramón Zatarain Cabada, María Lucía Barrón Estrada,
Francisco González Hernández, Raúl Oramas Bustillos

Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,
México

{rzatarain, lbarron}@itculiacan.edu.mx

Resumen. En este artículo se presenta un ambiente inteligente de aprendizaje con manejo afectivo para Java llamado Java Sensei. El sistema está diseñado para ayudar a los estudiantes de programación a reforzar distintas áreas del conocimiento sobre Java. El sistema evalúa aspectos como el estado cognitivo y afectivo del estudiante para tomar las estrategias de intervención que realizara agente pedagógico, así como realiza procesos de adaptabilidad por medio de un sistema de recomendaciones. El manejo pedagógico se realizó con una perspectiva Example-Tracing con manejo afectivo.

Palabras clave: entornos inteligentes de aprendizaje, sistemas tutores inteligentes, computación afectiva, lenguajes de programación, redes neuronales, sistemas lógicos difusos, sistemas de recomendación.

1. Introducción

La enseñanza y aprendizaje de la programación ha sido un reto complejo para todas las universidades en cualquier país. Algunos factores que contribuyen a su complejidad son: los métodos de enseñanza empleados por los instructores, los métodos de estudio empleados por los estudiantes así como sus habilidades y actitudes, el proceso de aprendizaje único que tiene la programación y la influencia psicológica de parte de la sociedad hacia los estudiantes [1, 2, 3, 4].

Algunas investigaciones han propuesto y desarrollado metodologías y herramientas que ayudan a los estudiantes a aprender como programar. Por ejemplo la ACM tiene el Grupo de Interés Especial sobre la Educación en Ciencias Computacionales (SIGCSE por sus siglas en inglés) donde se discuten problemas relacionados al desarrollo, implementación y/o evaluación de programas y planes de estudio [5].

Hoy en día en el campo de la inteligencia artificial, los Sistemas Tutores Inteligentes (ITS por sus siglas en inglés) intentan simular a un tutor humano proveyendo instrucciones personalizadas que toman en cuenta no solo aspectos cognitivos del estudiante [6] si no también elementos afectivos [7].

Este artículo presenta la implementación de un Entorno Inteligente de Aprendizaje (ILE por sus siglas en inglés) para el aprendizaje del lenguaje de programación Java. Esta herramienta es nombrada *Java Sensei*, la cual intenta crear un entorno flexible e interactivo que considere los estados afectivos y cognitivos de los estudiantes. La organización del artículo es la siguiente: en la sección 2, describimos el diseño e implementación del sistema con su arquitectura, componentes y algoritmos. Los resultados y pruebas de la aplicación se muestran en la sección 3 y las conclusiones y trabajos futuros en la sección 4.

2. Diseño e implementación de Java Sensei

El ILE *Java Sensei* consiste de siete componentes: *Plataforma Afectiva*, *Módulo del Estudiante o Usuario*, *Agente Pedagógico*, *Módulo de Adaptación*, *Módulo de Enseñanza (Pedagógico)*, *Módulo del Experto* y *el Motor ITS (Sistema de Tutoría Inteligente/Afectivo)*.

2.1. Plataforma afectiva

En el módulo afectivo se creó una plataforma emocional basada en el ITS AutoTutor [9] donde se implementó un sistema lógico difuso y una red neuronal de propagación hacia atrás para el reconocimiento de emociones. La plataforma emocional representa un modelado tanto del tutor como del estudiante. Ambos componen las variables de entrada y salida que necesita el sistema lógico difuso. Estos modelos y variables son descritos a continuación.

Modelo del Estudiante: El modelo del estudiante representa la parte cognitiva y emocional del estudiante. Este consiste de dos aspectos cognitivos y dos emocionales explicadas en la Tabla 1.

Tabla 1. Variables de entrada que representan el modelo del estudiante.

Nombre de la variable de entrada	Definición
Emoción Actual	Es la emoción actual del estudiante cuando responde un ejercicio. La emoción es obtenida a través de la aplicación de reconocimiento emocional. Los posibles valores son: Feliz, Enojado, Triste, Sorprendido y Neutral.
Emoción Previa	Es la emoción previa del estudiante. Sus valores son del mismo tipo que en la Emoción Actual.
Habilidad Global del Estudiante	Es la eficiencia que posee el estudiante con respecto al curso en su totalidad.
Calidad de la Respuesta Actual	Es la eficiencia que tiene el estudiante con respecto al ejercicio que está resolviendo.

Acciones del Tutor: Las acciones del tutor representan la respuesta emocional de un tutor humano y están representadas como variables de salidas del sistema lógico difuso que son usadas por el componente del *Agente Pedagógico* y el *Motor ITS*. Ambos serán explicados más adelante. Las variables están definidas en la Tabla 2.

Tabla 2. Variables que representan las acciones del tutor.

Nombre de la variable de entrada	Definición
Retroalimentación	Este es un mensaje que da una perspectiva positiva, neutral o negativa del estudiante con respecto a su habilidad global y la calidad de sus respuestas (ambas variables mencionadas anteriormente). Representa la retroalimentación que daría un tutor humano cuando él sabe si el progreso académico es correcto o no.
Respuesta empática emocional	Son frases recibidas por el estudiante de parte del tutor estas dependerán de los estados emocionales presentados por el estudiante. Están categorizadas en: neutral, encantado, sorprendido, compasivo y escéptico
Expresión Facial	Es la expresión facial que tendrá que usar el Agente Pedagógico. Categorizadas en Encantado, Escéptico, Sorprendido y Neutral.
Intervención	Este valor representa si el agente pedagógico debe realizar una intervención (aparecer en pantalla) al estudiante.

2.2. Sistema lógico difuso y red neuronal

El sistema lógico difuso busca representar la forma en la que un tutor maneja varios escenarios relacionados con la situación emocional y cognitiva de un estudiante [10]. Estas situaciones tienen que hacerse con acciones predefinidas.

El sistema lógico difuso usa los valores emocionales obtenidos de la red neuronal la cual reconoce los estados afectivos del estudiante cuando este trabaja con el ILE *Java Sensei*. El sistema lógico difuso trabaja con cuatro variables de entrada difusa y tres de salida. Cuenta con un total de 144 reglas difusas construidas. Para definir esas reglas se usó el Lenguaje de Control Difuso, un lenguaje estandarizado para la creación de reglas difusas. La implementación fue hecha con *jFuzzyLogic* [11].

El sistema de reconocimiento emocional fue construido en tres pasos: El primer paso fue una implementación de un extractor de características de imágenes faciales. En el segundo paso se usó este extractor sobre un corpus de datos para poder obtener los datos y entrenar la red neuronal. Usamos algoritmos basados en Java implementados en NeuroPH [21] para realizar el entrenamiento, el clasificador usado en la red neuronal fue propagación del error hacia atrás. En el tercer paso se integró la red neuronal con el sistema lógico difuso. Ambos son parte del ILE. Para el entrenamiento y prueba de la red neuronal usamos el corpus RAFD (Radboud Faces Database), el cual es una base de datos con 8040 expresiones faciales diferentes que contienen un conjunto de 67 modelos incluyendo hombres y mujeres. Una vez que el estado emocional es extraído del estudiante, este es enviado al sistema lógico difuso. La Fig. 1 muestra los pasos seguidos en el reconocimiento y retroalimentación afectivos del sistema. En el primer paso del ILE este recibe los datos y colección de imágenes del estudiante desde el navegador. En el segundo paso el ILE inicia un ciclo en el cual las imágenes del rostro del estudiante son enviadas a la red neuronal y decide al final del ciclo la emoción actual. Esta emoción actual es obtenida basándose en la emoción más repetida del estudiante. En el tercer paso el ILE envía el modelo del estudiante actualizado al sistema lógico difuso el cual ejecuta las reglas difusas para obtener las salidas de las acciones del tutor. En este último paso, las acciones del tutor son enviadas al navegador para construir las acciones del *Agente Pedagógico*.

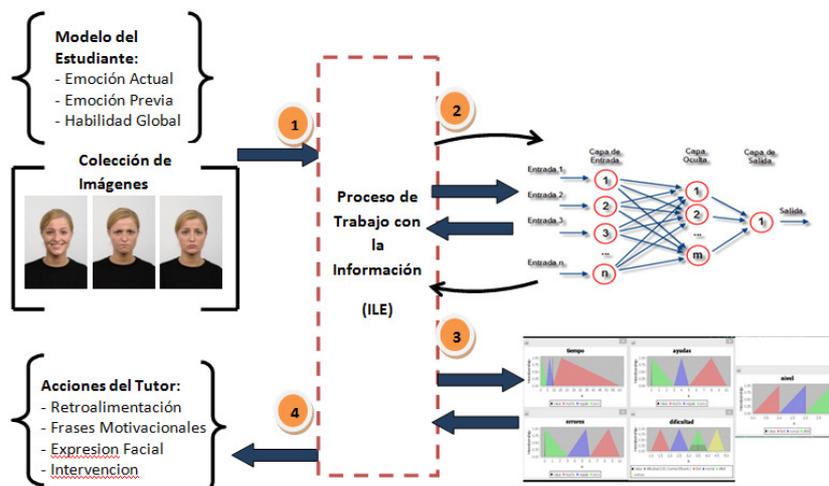


Fig. 1. Pasos seguidos por el sistema lógico difuso y la red neuronal.

2.3. Agente pedagógico

El agente pedagógico es responsable de transmitir los mensajes y expresiones que un tutor humano debería realizar. Para esto se le ha provisto con expresiones faciales y diálogos con los que pueda comunicarse con el estudiante. Esas expresiones están

basadas en las expresiones faciales investigadas por Ekman [12]. La posición de las cejas, boca y apertura de los ojos son tomadas en cuenta. Estas posiciones se utilizan como entrada en el programa FaceGen Modeller Free [13]. En la Fig. 2 se observan las cuatro expresiones faciales hechas para el agente pedagógico de *Java Sensei*. Estas expresiones están basadas en el trabajo realizado con el agente pedagógico de Auto-Tutor [10].

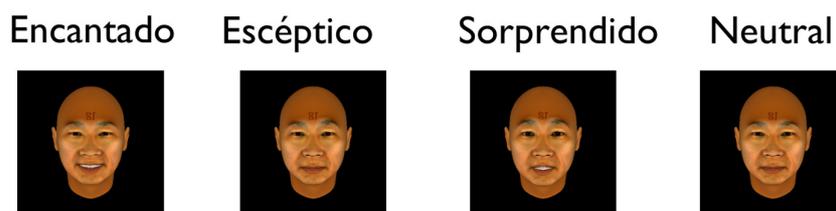


Fig. 2. Expresiones faciales del *Agente Pedagógico*.

2.4. Módulo de adaptación

Una característica de los ILE es que estos deben poder adaptarse al estudiante [8]. *Java Sensei* colecciona los puntajes producidos por los estudiantes al visitar los recursos o finalizar los ejercicios con los que cuenta el ILE. El puntaje de los ejercicios se mide usando la escala de Likert [14] donde el valor de uno indica que hay un desagrado total y el valor cinco indica que hay un agrado total. Cuando el estudiante ingresa por primera vez al sistema todos los ejercicios se inicializan con un valor de dos, con esto se coloca un valor que represente neutralidad en el gusto del estudiante. Para poder utilizar estos puntajes en la adaptabilidad creamos dos sistemas de recomendación uno para ejercicios y otro para recursos usando Apache Mahout [15]. El sistema implementa el enfoque K-Nearest Neighbor (k-NN) y la Correlación de Pearson. Entre más usuarios participen en el sistema, se generaran mejores recomendaciones y de este modo se reflejara una mejor adaptación hacia cada usuario.

2.5. Módulo pedagógico

El sistema usa un modelo pedagógico conocido como Problem-solving [6] en este los estudiantes aprenden resolviendo problemas los cuales cuentan con una cierta estructura. El sistema usa tres tipos de estrategias para solucionar los problemas. El tipo 1 es usado para evaluar conceptos teóricos con ejercicios de opción falso y verdadero. El tipo 2 presenta piezas de código que el estudiante tendrá que seleccionar para completar un programa completo y generar una salida deseada. El tipo 3 es una combinación de los dos tipos previos y permite crear ejercicios más complejos que involucren “n” número de pasos para alcanzar la solución.

2.6. Módulo del experto

Este modelo contiene el conocimiento del experto que el estudiante quiere aprender y practicar. El sistema representa el conocimiento usando archivos JSON (una

variación de los archivos XML). La representación del conocimiento usa la teoría de espacios del conocimiento [19]. El modelo del experto se representa en seis habilidad básicas que el estudiante deberá dominar para así ser este modelado por un gráfico que representa el espacio del conocimiento. Estas habilidades son: Introducción a Java, Variables y Cálculos, Estructuras de Selección, Estructuras de Repetición, Métodos y Arreglos.

2.7. Motor ITS

Un ILE combina técnicas hipermedia con otros métodos usados por los ITS. Se aplicó la teoría Example-Tracing [16,17] con el fin de implementar la parte ITS en *Java Sensei*. Los Example-Tracing ITS tienen la ventaja de proveer paso por paso una guía para los estudiantes y proveer múltiples estrategias para poder resolver un problema, esto incluye manejo soluciones óptimas y sub-óptimas como conceptos erróneos comunes. Los ejercicios Example-Tracing pueden ser fácilmente adaptados con el modelo pedagógico Problem-solving (Resolución de Problemas) y ser representados por un grafo que es atravesado por el estudiante.

Una de las principales contribuciones de este trabajo es la integración de información cognitiva y emocional en un grafo de comportamiento el cual es usado por el agente pedagógico para mostrar emoción y empatía hacia el estudiante. Con esto se crea un concepto llamado Affective Example-Tracing. La Fig. 3 muestra parte de un grafo de comportamiento.

A continuación, se describe el algoritmo que se implementó en el Affective Example-Tracing [18]:

- a) El ILE carga el archivo JSON del ejercicio que se va a resolver.
- b) Dentro del archivo JSON, un grafo de comportamiento es creado y el paso inicial es ejecutado.
- c) Los estudiantes navegan por el gráfico (a través de disparos de eventos como el clic del ratón), teniendo diferentes opciones que puedes realizar y diferentes caminos a elegir. Las opciones son:
 - i. Paso Inicial (PI)
 - ii. Paso Erróneo (PE)
 - iii. Paso Optimo (PO)
 - iv. Paso Sub-Optimo (PSO)
 - v. Paso Final Optimo (PFO)
 - vi. Paso Final Sub-Optimo (PFSO)
- d) Cada paso lanza dos eventos:
 - i. Evaluación de la emoción actual.
 - ii. Evaluación de la acción del tutor.

Cada vez que un estudiante escoge alguna de las opciones en el grafo, el motor llama al sistema lógico difuso para obtener las acciones del tutor (mensajes de retroalimentación, realizar una intervención, etc.) y así obtener la información emocional para que el *Agente Pedagógico* pueda construir su intervención y el *Motor ITS* construir la pertinente representación gráfica. En la figura 4 se ilustra un ejemplo en el

cual se observa como de lado izquierdo de la figura los estudiantes resuelven un problema en un camino óptimo y de lado derecho el estudiante llega a un camino erróneo.

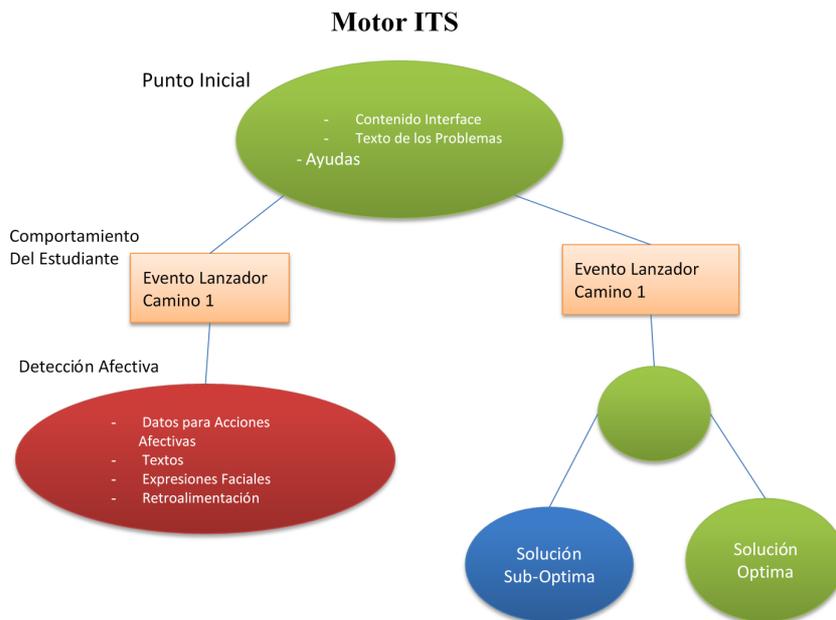


Fig. 3. Ejemplo de grafo de comportamiento.

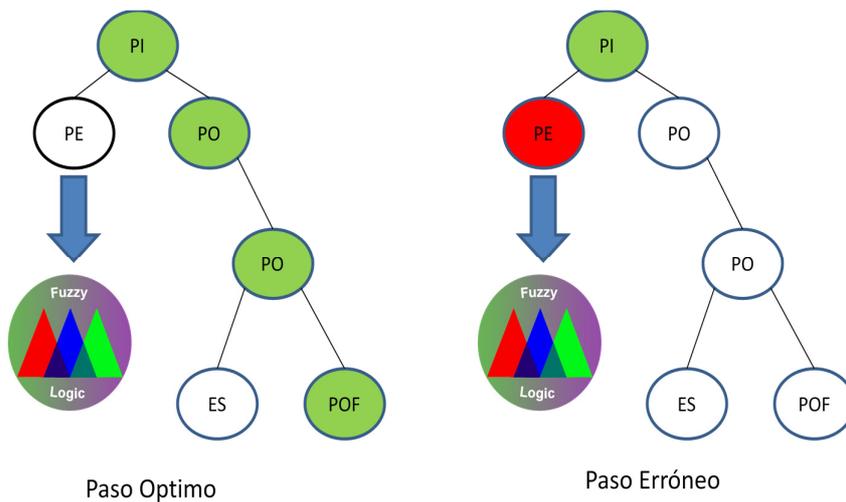


Fig. 4. Ejemplo de navegación a través del árbol.

3. Resultados y comparaciones

Para evaluar el ILE Java Sensei se consideran dos aspectos: la funcionalidad del software y la utilización del sistema con un grupo de estudiantes. Para evaluar la funcionalidad del software se comparó con otros tres sistemas similares al ILE Java-Sensei. Estos sistemas son JavaTutor System [23], JITS [20] y jLatte [24]. Para este trabajo, hemos decidido evaluar nueve características de un ILE (ITS) [6] utilizando una matriz ponderada utilizando la escala de Likert [14] de 1 a 5, dónde 1 representa una característica incompleta y 5 una característica completa. La Tabla 3 muestra los resultados obtenidos de la evaluación de la funcionalidad del software.

Tabla 3. Tareas realizadas en la fase de pruebas.

Característica	JavaSensei	JavaTutor	JITS	jLatte
Generatividad	3	5	4	4
Conocimiento experto y del estudiante	5	5	5	5
Iniciativa mixta	2	5	1	1
Aprendizaje Interactivo	4	5	4	5
Modelo instruccional	5	5	5	5
Auto-mejora	1	1	1	1
Manejo de emociones	4	5	1	1
Sistema web	5	5	1	1
Sistema móvil	5	1	1	1
Totales:	34	37	23	24

Los sistemas evaluados cumplen con casi todas las características de un sistema ILE/ITS a excepción de las características 3 (iniciativa mixta) y 6 (auto-mejora). La característica 7 (el manejo de emociones) es un área reciente por lo tanto son pocos los sistemas que incorporan esta característica. Lo mismo para las características 8 y 9. Se espera que un futuro los ITS desarrollados en los últimos años migren como sistemas Web y proporcionen soporte para dispositivos móviles.

La evaluación del uso del sistema se llevó a cabo una prueba en el Instituto Tecnológico de Culiacán, con el grupo ESA de 8:00 a 9:00 A.M., de la asignatura Programación Orientada a Objetos (POO), en donde se buscó medir el impacto del ILE Java Sensei en el tema de herencia en la POO. La selección de los estudiantes no se realizó de manera aleatoria, sino por disponibilidad de los participantes. Para acceder al ILE, los estudiantes utilizaron un navegador web desde con soporte de HTML 5 y que cuente con cámara frontal esta podría ser una computadora de escritorio, laptop o teléfono inteligente, ingresando al sistema desde su cuenta de Facebook.

Siguiendo una de las metodología propuesta por Woolf [6] para la evaluación del ITS se diseñó una prueba donde se aplicaría un Pre-Test (Examen Diagnostico), posteriormente habría una sesión de uso del programa y un examen Post-Test. Para esto primero se impartió una introducción al ILE Java Sensei al grupo anteriormente mencionado y al día siguiente se aplicó un examen diagnóstico para evaluar aspectos teó-

ricos y prácticos sobre: conceptos herencia, tipo de herencia, atributos, métodos heredados, palabra clave extends, herencia simple, diferencia entre métodos overload-override, final, static, polimorfismo y la utilización del método super().

Posteriormente se dividió el grupo en dos partes, dónde el Grupo 1 sería el grupo experimental y utilizaría el ILE JavaSensei y el Grupo 2 representaría el grupo de control. La tabla 4 resume las actividades realizadas.

Tabla 4. Tareas realizadas en la fase de pruebas.

Fase	Tarea	Fecha	Participantes	¡Latte
1	Introducción al Ambiente ILE JavaSensei	24/03/2015	20	20 mins.
2	Examen Diagnóstico	25/03/2015	20	16 mins.
3	Utilización ILE JavaSensei	26/03/2015	10	50 mins.
4	Examen Post-Test	27/03/2015	20	20 mins.

Una vez que el Grupo 1 estudió el tema de herencia utilizando el ILE JavaSensei se procedió a aplicar el examen Post-Test a ambos grupos.

Al realizar el análisis del examen diagnóstico se obtuvo un 80% de aprobados y el 20% restante de reprobados. De manera similar, con el examen Post-Test se obtuvo un 85% de aprobados con el correspondiente 15% de reprobados. La figura 5 muestra los resultados obtenidos.



Fig. 5. Resultados de los exámenes diagnóstico y post-test.

De la gráfica anterior podemos observar una mejoría del 5% antes y después de los exámenes. Al realizar el análisis comparativo del Grupo 1 que utilizaron el ILE JavaSensei (ver tabla 4), el 20% mejoró su puntaje respecto al examen diagnóstico, el 40% no mostro cambio alguno y el 40% disminuyo su calificación.

4. Conclusiones y trabajo a futuro

Dados los resultados de las pruebas estadísticas realizadas, se puede considerar que la utilización de un Ambiente ILE tiene un efecto positivo en el aprendizaje de los

estudiantes. La variación negativa fue menor por lo que un ILE es una herramienta que ayuda a reforzar los distintos temas que el estudiante pueda estar cursando en su curso de programación, con las ventajas de la disponibilidad que un sistema web tiene.

Se espera a futuro poder adaptar un Módulo de colaboración social al ILE [22] ya que además de las emociones se ha comprobado la importancia de la interacción social de los estudiante, ya que esto lo ayuda a obtener herramientas para poder enfrentar mejor los retos que “desconoce” y tener un lugar donde pueda debatir sus ideas con respecto a los ejercicios y/o lecciones que está enfrentando.

Referencias

1. Matthíasdóttir, Á.: How to teach programming languages to novice students? Lecturing or not. In: International Conference on Computer Systems and Technologies-CompSysTech. (2006)
2. Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (2002)
3. Gomes, A. and A.J. Mendes: Learning to program-difficulties and solutions. In: International Conference on Engineering Education-ICEE (2007)
4. Bernard, M. and Bachu, E.: Enhancing the Metacognitive Skill of Novice Programmers through Collaborative Learning. In: Metacognition: Fundamentals, Applications, and Trends. (2015), Springer. p. 277–298
5. SIGCSE (2015) [cited 2015]; Available from: <http://www.sigcse.org/>
6. Woolf, B.P.: Building Intelligent Interactive Tutors. Amherst, Massachusetts, Estados Unidos: Morgan Kaufmann Publishers (2009)
7. Picard, R.W.: Affective computing. MIT press (2000)
8. Brusilovsky, P. and E. Millán: User models for adaptive hypermedia and adaptive educational systems. In: The adaptive web. Springer-Verlag (2007)
9. D’Mello, S., R.W. Picard, and A. Graesser: Toward an affect-sensitive AutoTutor. *IEEE Intelligent Systems*, 4, pp. 53–61 (2007)
10. D’Mello, S., et al.: AutoTutor detects and responds to learners affective and cognitive states. In: Workshop on Emotional and Cognitive Issues at the International Conference on Intelligent Tutoring Systems (2008)
11. Cingolani, P. and J. Alcalá-Fdez: jFuzzyLogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming. *International Journal of Computational Intelligence Systems*, 6(sup1), pp. 61–75 (2013)
12. Ekman, P.: Facial Expressions. (1999)
13. Inversions, S.: FaceGen modeller: (Version 3.3) [computer software]. Toronto, ON: Singular Inversions (2008)
14. Likert, R.. A method of constructing an attitude scale. *Scaling: a sourcebook for behavioral scientists*. Chicago: Aldine, pp. 233–243 (1974)
15. Owen, S., et al.: Mahout in action. (2011)
16. Aleven, V., et al.: A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2), pp. 105–154 (2009)
17. Aleven, V., et al.: The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains. In: *Intelligent Tutoring Systems*, Springer (2006)
18. Kumar, R., et al.: Comparison of Algorithms for Automatically Building Example-Tracing Tutor Models. In: *Educational Data Mining* (2014)
19. Doignon, J. P., & Falzague, J. C.: Knowledge spaces. Springer (1999)

20. Wallis, M.: JavaTutor-A Remotely Collaborative, Real-Time Distributed Intelligent Tutoring System for Introductory Java Computer Programming. A Qualitative Analysis (2011)
21. Bruxella, J. Marv Dallfin, S. Sadhana, and S. Geetha: Categorization of Data Mining Tools Based on Their Types. *International Journal of Computer Science and Mobile Computing*, 3(3), pp. 445–452 (2014)
22. Soller, A.: Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12, pp. 40–62 (2001)
23. Grafsgaard, J. F., Wiggins, J. B., Vail, A. K., Boyer, K. E., Wiebe, E. N., & Lester, J. C.: The Additive Value of Multimodal Features for Predicting Engagement, Frustration, and Learning during Tutoring. In: *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 42–49, ACM (2014)
24. Holland, J., Mitrovic, A., & Martin, B.: J-LATTE: a Constraint-based Tutor for Java. (2009)

Ambiente inteligente y afectivo de aprendizaje para compiladores

María Lucia Barrón-Estrada, Ramón Zatarain-Cabada, H.E. Gómez-Pérez

Instituto Tecnológico de Culiacán,
Culiacán, Sinaloa, México

{rzatarain, lbarron, hgomez}@itculiacan.edu.mx

Resumen. Este artículo presenta el diseño e implementación de un Ambiente Inteligente de Aprendizaje (AIA) capaz de considerar el perfil completo de un estudiante (incluyendo su estado emocional), para la enseñanza de un tema tan complicado como lo es compiladores. El AIA se encuentra enfocado en el método de *aprendiz* (Apprenticeship) el cual permite a los estudiantes aprender mediante la práctica, cambiando el arraigado paradigma del aprendizaje meramente teórico. Se incorporan características que permiten al AIA elegir el tipo de ejercicio que mejor convenga al estudiante, comenzando por los más sencillos e incrementando la dificultad, haciendo uso de clasificación con lógica difusa y reconocimiento de emociones.

Palabras clave: computación afectiva, sistemas tutores inteligentes, redes neuronales, sistemas difusos, aprendizaje máquina, ambientes inteligentes.

1. Introducción

Un ambiente inteligente de aprendizaje (AIA) es un nuevo tipo de sistema educacional, el cuál combina lo mejor de los sistemas tutores inteligentes (STI) y los ambientes de aprendizaje. Se ha comprobado la utilidad de los AIA para la enseñanza meramente práctica, poniendo a prueba a los estudiantes en la resolución de problemas reales y desafiantes.

En los últimos años los AIA han incorporado la habilidad para reconocer el estado afectivo de los estudiantes y, con ello, ser capaces de cambiar la manera de interactuar [1-5]. La detección de emociones en los usuarios en muchos trabajos se realiza utilizando sensores especiales como sillas de postura o brazaletes [6]. Estos dispositivos resultan intrusivos y, a veces, molestos para los usuarios. Tratando de evitar esta incomodidad al usuario, se han realizado trabajos en donde incluye la detección de emociones dentro del tutor [7].

En el presente trabajo se busca integrar en el AIA diferentes aspectos como son el estado emocional, nivel de conocimientos en compiladores, además de la metodología del *aprendiz* (apprenticeship) para la enseñanza [8]. Esto se logra mediante el componente neurodifusor, que trabaja con una API externa fuera del entorno del AIA, para el reconocimiento de emociones, además utiliza reglas y conjuntos difusos para

detectar el nivel del estudiante en base a la resolución de los ejercicios. La interfaz ingresa al estudiante directamente a la práctica y de acuerdo al nivel detectado propone la complejidad de los ejercicios, simulando así un experto con su discípulo.

Un AIA es tan inteligente como el conocimiento introducido por sus expertos, por lo que al implementar una plataforma que sea fácilmente extensible por múltiples especialistas, obtendrá experiencia que le permitirá igualar e incluso superar a cualquier otro AIA. Es por ello que mostramos resultados haciendo comparación con diversos tutores [9-11] que son considerados los mejores en sus respectivas áreas.

El artículo está dividido de la siguiente manera: la sección 2, *Dominio del AIA* explica los distintos objetivos y habilidades que el estudiante debe adquirir al utilizar el sistema. La sección 3, *Interfaz del AIA* describe las distintas áreas que conforman la interfaz y como cada una cumple su propósito. La sección 4, *Arquitectura del AIA* muestra la estructura y relación de los componentes más importantes, haciendo énfasis en los componentes que manejan la parte de inteligencia artificial y afecto. La sección 5, *Trabajo relacionado* presenta la descripción de otros trabajos y los diversos aspectos que fueron seleccionados para realizar una comparación entre el sistema propuesto y los otros existentes. Por último las *Conclusiones* expresan los resultados obtenidos y el trabajo que se planea para el futuro.

2. Dominio del ambiente inteligente de aprendizaje

Compiladores es un microcosmo del área de ciencias de la computación, ya que contiene problemas de todos los tipos incluyendo entre otros (pero no limitado a) inteligencia artificial (búsquedas heurísticas), algoritmos, lenguajes formales, autómatas y arquitectura [12].

El ambiente inteligente de aprendizaje busca cumplir objetivos específicos, limitando el amplio panorama teórico y enfocándose en las habilidades prácticas que requiere el aprendiz. Para esto, el AIA abarca diferentes temas que el estudiante debe cubrir en una o más sesiones, alcanzando los objetivos siguientes:

- Comprende la notación Backus-Naur (BNF).
- Crea un lenguaje en BNF y puede escribir en él.
- Relaciona un conjunto de símbolos y los agrupa (tokens).
- Genera una interfaz de uso común para la clasificación de tokens.
- Comprende la función de un scanner.
- Crea un scanner en java utilizando la interfaz de clasificación de tokens.
- Comprende la función de un parser.
- Genera un parser utilizando el scanner previo.

Todos estos objetivos son enseñados con las mejores prácticas, de esta manera el estudiante desarrollará un compilador básico y adquirirá las habilidades prácticas que se requieren en la materia.

3. Interfaz del ambiente inteligente de aprendizaje

El AIA cuenta con una interfaz diseñada responsivamente, la cual se adapta al tamaño de pantalla del dispositivo en el que se esté utilizando. En la Fig. 1 se puede observar el área principal del AIA, donde el estudiante aprende mediante la práctica.

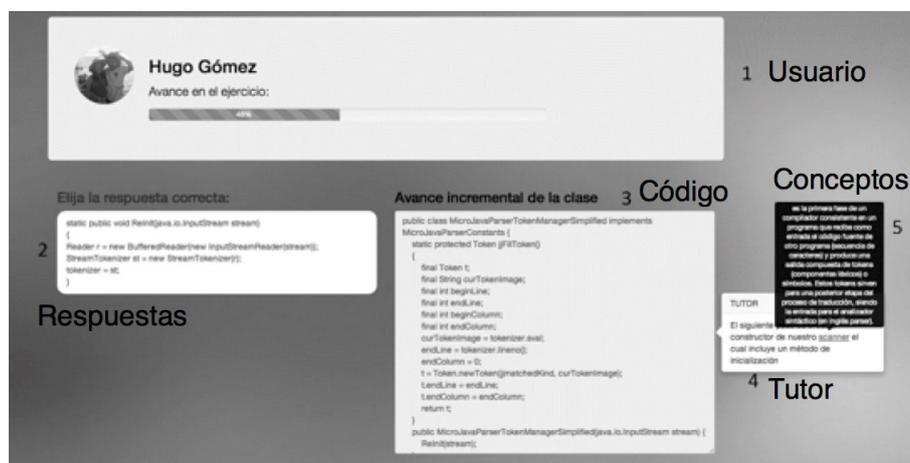


Fig. 1. Interfaz en el área de ejercicios del AIA.

La interfaz se encuentra organizada en cinco secciones delimitadas por áreas físicas, en donde cada una tiene un objetivo específico. Las áreas son las siguientes:

1. **Usuario.** Presenta los datos del estudiante, mostrando su foto de perfil de Facebook y el avance del ejercicio actual.
2. **Respuestas.** El área de respuestas es donde influye la detección inteligente del ambiente, ya que dependiendo de la clasificación del estudiante, estas cambian de complejidad. Existen tres niveles de complejidad para las respuestas solicitadas. El primer nivel muestra una parte del código del compilador que el estudiante tiene que teclear tal cual. El segundo nivel muestra una variedad de respuestas, de las cuales el estudiante tiene que elegir la mejor. La tercera y más desafiante, muestra código incompleto, el cual tiene que ser completado por el estudiante usando sus habilidades de intuición, así como conocimientos previos en la materia.
3. **Código.** Aquí se muestra el avance del código que el estudiante ha completado, de esta manera puede analizar lo que ha realizado y darse una mejor idea de cómo los pequeños avances poco a poco toman forma, para terminar una parte del compilador.

4. **Tutor.** Esta zona muestra el conocimiento del experto, donde explica que es lo que está pasando conforme se avanza bloque a bloque de código, de manera que el estudiante vea cuáles son las mejores prácticas y alternativas con las que se pueden realizar los mismos objetivos.
5. **Conceptos.** Además de la explicación del experto, las palabras clave se encuentran resaltadas para que el estudiante pueda ver sus definiciones sin necesidad de salir del AIA.

4. Arquitectura del ambiente inteligente de aprendizaje

El AIA cuenta con una arquitectura de capas relajadas que se puede visualizar en la Fig. 2. A continuación se detallan los componentes más relevantes de cada capa.

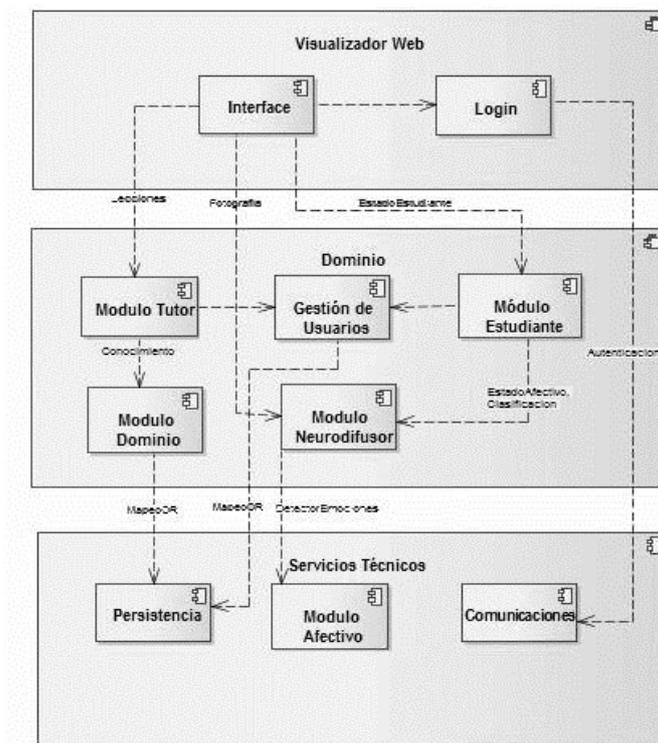


Fig. 2. Arquitectura del ambiente inteligente de aprendizaje.

4.1. Capa de visualizador Web

Es utilizada para la presentación del AIA y es con lo que interactúa el usuario. Esta cuenta con dos componentes llamados Interface y Login.

4.1.1. Interface

Se encarga de presentar la interfaz al usuario y está hecha mediante HTML5, utilizando el framework Bootstrap de twitter para conseguir un diseño responsivo capaz de adaptarse a cualquier tamaño de visualización.

4.1.2. Login

Provee la comunicación para ingresar al tutor por medio de Facebook actualmente, pero se planea acrecentarlo a google+ y twitter.

4.2. Capa de dominio

La capa de dominio implementa todo lo relacionado con el conocimiento del AIA, inteligencia artificial y pase de la información hacia el almacenamiento, todo en esta capa está implementado en PHP, mediante el framework Laravel y consta de cinco componentes que se describen a continuación.

4.2.1. Módulo tutor

Codifica los métodos de enseñanza que sean adecuados para el área de estudio y el alumno. Basado en los datos de las debilidades y fortalezas, nivel de conocimiento y emoción, el AIA selecciona el modelo de intervención de enseñanza más apropiado. Por ejemplo, si un estudiante ha sido evaluado como principiante en un determinado ejercicio, el módulo muestra algunos ejemplos paso a paso del procedimiento antes de invitar al usuario a realizarlo el mismo. También puede proporcionar retroalimentación, explicaciones y el entrenamiento a medida que el participante realiza el proceso. Conforme el estudiante gana experiencia, el módulo puede decidir presentar los escenarios cada vez más complejos. También puede decidir tomar el asiento de atrás y dejar que la persona explore libremente los ejercicios, intervenir con las explicaciones y el entrenamiento sólo bajo petición. Además, el módulo se encarga de poder elegir temas y ejemplos que subsanen las deficiencias de competencia del usuario.

4.2.2. Módulo dominio

Contiene una representación computacional de un experto en el área de Compiladores. Este conocimiento permite al AIA comparar las acciones del estudiante y las selecciones con las de un experto a fin de evaluar lo que el usuario conoce y lo que necesita aprender. Por ejemplo, se capta el conocimiento del estudiante a base de reglas que permite que el AIA genere problemas sobre la marcha.

Además permite que el desarrollador del curso pueda crear plantillas, que especifican una secuencia permitida de acciones correctas. Este método evita la codificación de la capacidad de resolver todos los problemas posibles. En cambio, sólo requiere la capacidad de especificar cómo el alumno deberá responder en cada

escenario. Estas plantillas se encuentran implementadas en formato JSON para su fácil manejo.

4.2.3. Módulo gestor de usuarios

Es el componente que accede la capa de servicios para que la persistencia de la información sea totalmente encapsulada y el acceso se vea comprensible y simple.

4.2.4. Módulo estudiante

Evalúa el desempeño de cada alumno para determinar sus conocimientos, habilidades perceptivas y habilidades de razonamiento. Al mantener y hacer referencia a un módulo con el detalle de las fortalezas y las debilidades de los usuarios, el AIA puede ofrecer muy específicamente, la instrucción pertinente.

4.2.5. Módulo neurodifusor

Contiene a las clases necesarias para la lógica difusa y la conexión hacia el componente afectivo, que proporcionan la información necesaria para que el AIA tome las decisiones correspondientes en base a estas técnicas de inteligencia artificial.

Lógica difusa

El STI utiliza la teoría de conjuntos difusos y reglas difusas para calibrar lo vago o ambiguo en las 4 variables difusas que maneja, como son tiempo, emoción, nivel en el ejercicio, como variables de entrada y la cuarta es la variable de salida, llamada nivel de dificultad, esta será tomada en cuenta para el próximo ejercicio que se le presentará al estudiante.

Un conjunto difuso puede ser definido simplemente como un conjunto con límites difusos [13].

Visto de otra manera, en la teoría difusa, un conjunto difuso A de universo X esta definido por la función $\mu_A(x)$ llamada función de membresía del conjunto A la cual es representada como: $\mu_A(x): X \rightarrow [0, 1]$ donde:

- $\mu_A(x) = 1$ si x esta totalmente en A ;
- $\mu_A(x) = 0$ si x no esta en A ;
- $0 < \mu_A(x) < 1$ si x esta parcialmente en A .

Una regla difusa puede ser definida como un enunciado condicional con variables lingüísticas y valores lingüísticos determinados por los conjuntos difusos en el universo del discurso.

Algunas reglas difusas que se manejan en el AIA son las siguientes:

1. **If** (nivelEnEjercicio is bajo) **and** (tiempoResolviendo is lento) **and** (estadoAfectivo is enganchado) **then** (dificultadEjercicio is baja)
2. **If** (nivelEnEjercicio is regular) **and** (tiempoResolviendo is rapido) **and** (estadoAfectivo is distraido) **then** (dificultadEjercicio is alta)
3. **If** (nivelEnEjercicio is regular) **and** (tiempoResolviendo is lento) **and** (estadoAfectivo is neutral) **then** (dificultadEjercicio is baja)
4. **If** (nivelEnEjercicio is regular) **and** (tiempoResolviendo is regular) **and** (estadoAfectivo is enganchado) **then** (dificultadEjercicio is alta)

5. **If** (nivelEnEjercicio is bueno) **and** (tiempoResolviendo is rapido) **and** (estadoAfectivo is enganchado)
then (dificultadEjercicio is alta)

4.3. Capa servicios técnicos

Contiene todos los servicios de bajo nivel, como lo son base de datos y comunicación entre sistemas y componentes de detección de emociones. Esta capa cuenta con tres componentes que se describen a continuación.

4.3.1. Módulo afectivo

Se encarga de procesar las imágenes y devolver la emoción que se detectó hacia la capa superior, esto lo hace mediante una API externa de Sky Biometry la cual utiliza algoritmos de aprendizaje máquina y una inmensa base de datos con millones de caras de alrededor del mundo, para identificar las coordenadas de la boca, ojos y nariz y así determinar la emoción actual [14].

4.3.2. Persistencia

Maneja todo lo relacionado con base de datos, debido a que en ninguna otra parte del AIA existen accesos para guardar la información. Este componente provee las interfaces necesarias para el guardado y al utilizar tecnología ORM mediante el framework laravel este puede guardar para distintos proveedores.

4.3.3. Comunicaciones

Se utiliza para la interacción con otros sistemas, como lo es el log in para Facebook. Además es el componente que provee la conexión hacia el compilador en un servidor apache tomcat, permitiendo realizar los ejercicios de comprobación del lenguaje BNF.

5. Trabajo relacionado

Se decidió realizar una evaluación comparativa con otros sistemas que tienen una finalidad similar [9-11], la cual se puede observar en la Tabla 1. En esta se listan diversas características de evaluación propuestas por Woolf [5] y se muestra una escala de Likert con valores de 5 a 1 representando valores desde Totalmente de acuerdo hasta totalmente en desacuerdo respectivamente.

Los sistemas con los que se hace la comparación son ambientes inteligentes de aprendizaje diseñados para otro dominio (programación de computadoras) debido a la falta de sistemas de aprendizaje en el área de compiladores.

El ambiente inteligente para el aprendizaje de introducción a la programación (Intelligent Environment for Learning Introductory Programming (ITEM/IP)) [11]

soporta un curso de introducción a la programación basado en el mini lenguaje turingal, como base para dominar los conceptos principales de la programación.

NoobLab [9] trata de replicar el dinamismo que existe entre el tutor y el estudiante dentro de los talleres de trabajo para programación, en un ambiente totalmente en línea.

Web-Based Intelligent Tutoring System for Computer Programming (WITS) [10] se encuentra dirigido por un proceso de decisiones realizado a partir de redes bayesianas, las cuales permiten recomendar metas de aprendizaje y generar secuencias de lectura.

Tabla 1. Evaluación comparativa del AIA.

Característica del Tutor Inteligente	Descripción de la característica	ITEM/ITEM P	NoobLab	WITS	AIA
Generatividad	Capacidad de generar problemas apropiados, pistas y ayuda necesaria, adaptados a las necesidades de aprendizaje del estudiante.	3	5	4	5
Modelo del estudiante	Capacidad de representar y razonar acerca del conocimiento del estudiante y las necesidades de aprendizaje en respuesta a la proporción de instrucciones.	3	5	4	5
Modelo del experto	Representación y ruta de razonamiento acerca del rendimiento dentro del dominio y la capacidad implícita de respuesta por instrucción proporcionada.	5	5	3	5
Iniciativa Mixta	Capacidad de iniciar la interacción con el estudiante, así como la interpretación y respuesta de manera útil a interacciones iniciadas por los estudiantes.	5	3	4	4
Aprendizaje interactivo	Actividades de aprendizaje que requieren compromiso y autenticidad contextualizados en el dominio apropiado.	5	4	4	3
Modelo instruccional	Capacidad de cambiar el modo de enseñanza basado en inferencias sobre el aprendizaje del alumno.	4	4	4	4
Auto-mejora	Capacidad del sistema para monitorear, evaluar y mejorar su propia enseñanza basado en su experiencia con otros alumnos.	3	3	3	2

6. Conclusiones

Durante el desarrollo se buscó implementar todas las características que requiere un buen AIA, estas fueron: reconocimiento afectivo, reacción inteligente hacia las necesidades cognitivas del estudiante y una metodología de enseñanza eficiente.

Distintas metodologías fueron evaluadas, pero resultó triunfante la del *aprendiz* (apprenticeship) ya que promueve el uso de la práctica e interacción continua con el AIA. Un desafío muy grande fue la implementación de la interfaz para distintos dispositivos, que conservara la filosofía del *aprendiz*, esto fue resuelto mediante un diseño responsivo web, que permite el acceso a cualquier persona de todo el mundo.

Otra de las aportaciones valiosas en el proyecto fue la implementación del acceso a través de las redes sociales, las cuales son utilizadas día con día por un gran número de estudiantes, permitiendo esto, abrir un espacio para la participación del AIA. Además de que genera un gran impacto hacia estudiantes y profesores que requieren más ayuda para un tema tan complicado como lo es compiladores.

Hasta este momento, las pruebas realizadas al sistema permiten visualizar que el uso de un ambiente inteligente de aprendizaje para compiladores ha permitido que los estudiantes logren entender en un menor tiempo los temas referentes a análisis lexicográfico y análisis sintáctico en compiladores, al realizar de manera práctica algunos ejercicios desarrollando el código necesario para implementar la funcionalidad.

En trabajos futuros se pretende aplicar un mejor algoritmo de clasificación con redes bayesianas, agregar otras maneras de reconocer el estado cognitivo del estudiante y extender las lecciones para que el AIA realice compilación en tiempo de ejecución.

Referencias

1. Alevan, V., Koedinger, K.: An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26(2), pp. 147–179 (2002)
2. Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-aware tutors: recognising and responding to student affect. *Int. J. Learn. Technol.* 4, pp. 129–164 (2009)
3. McQuiggan, S.W., Lester, J.C.: Diagnosing self-efficacy in intelligent tutoring systems: an empirical study. In: *Proceedings of the 8th international conference on Intelligent Tutoring Systems*, pp. 565–574, Springer-Verlag, Jhongli, Taiwan (2006)
4. Im Jraidi, N., Chalfoun, P., Frasson, C.: Implicit strategies for intelligent tutoring systems. In: *Proceedings of the 11th international conference on Intelligent Tutoring Systems*, pp. 1–10, Springer-Verlag, Chania, Crete, Greece (2012)
5. Woolf, B. P.: *Building intelligent interactive tutors*. Morgan Kaufmann (2009)
6. Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: *Proceedings of the 14th international conference on artificial intelligence in education (AIED 2009)*, Brighton, UK, pp. 17–24, IOS press, Amsterdam (2009)
7. Zatarain-Cabada, R., Barrón-Estrada, M.L., Beltrán, J.A., Cibrian, F.L., Reyes-García, C., Hernández, Y.: Fermat: merging affective tutoring systems with learning social networks. In: *Proceedings of the 12th IEEE international conference on advanced learning technologies*, Rome, Italy, pp. 337–339, IEEE Computer Society (2012)
8. Patrick, A., Rainbird, H.: *Apprenticeship: Towards a New Paradigm of Learning*. Taylor & Francis (2014)
9. Neve, P.; Hunter, G.; Livingston, D.; Orwell, J.: NoobLab: An Intelligent Learning Environment for Teaching Programming. In: *Web Intelligence and Intelligent Agent*

- Technology (WI-IAT), IEEE/WIC/ACM International Conferences on, vol. 3, pp. 357–361 (2012)
10. Butz, C. J., et al.: A Web-Based Intelligent Tutoring System for Computer Programming. Web Intelligence, 2004. In: WI 2004, Proceedings of IEEE/WIC/ACM International Conference on (2004)
 11. Brusilovsky, P.: Towards an Intelligent Environment for Learning Introductory inProgramming. In: Cognitive Models and Intelligent Environments for Learning Programming. E. Lemut, B. du Boulay and G. Dettori, Springer Berlin Heidelberg. vol. 111, pp. 114–124 (1993)
 12. Appel, A. W.: Modern Compiler Implementation in Java. Cambridge. MA: Cambridge University Press (2002)
 13. Negnevitsky, M.: Artificial Intelligence a guide to intelligent systems. Pearson Education (2005)
 14. Sky Biometry: Cloud-based Face Detection and Recognition API. <https://www.skybiometry.com>. Accessed 15 Nov (2014)

An Intelligent Hospital Device for Health Caring Assistance of Patients with Motor Disabilities

Eduardo Vázquez-Santacruz, Mariano Gamboa-Zúñiga

CGSTIC, Cinvestav-IPN, Mexico, D.F., Mexico

{efvazquez,mgamboaz}@cinvestav.mx

<http://administracion.cinvestav.mx/SecretariadePlaneacion/CoordinacindeServiciosdeApoyo/CGSTIC.aspx>

Abstract. In this document we present an intelligence device which includes a monitor system for automatic movements of a robotic hospital bed based on posture classification and identification. This functionality has been developed in response to the requirements defined by the application of a diagnostic methodology [10]. Our diagnostic methodology applied on a Mexican public hospital allowed us to understand that the problem of mobility in elderly people and disabled people involves solutions like devices that attend these individuals to achieve movement without many complications. The movement of these patients can be performed routinely or sporadically during their stay in hospital. For patients who require certain routine application of this movement is designed an intelligent monitor system. This intelligent system allows medical experts to program the movements of the robotic bed considering the patient's posture and time. We will show a methodology for bodies classification in real time. Our work considers the problem of human posture classification with few information (sensors), applying digital process to expand the original data of the sensors and so get more significant data for the classification, however, this is done with low-cost algorithms to ensure the real-time execution.

Keywords: intelligent hospital devices, intelligent monitor, assistive devices, health care technology, applied artificial intelligence, posture recognition.

1 Introduction

In recent years the artificial intelligent systems (IS) have been used in several applications such as industrial control, robot control, traffic surveillance, remote sensing, and speech recognition to mention a few. In particular, IS applied to medical environments has been a challenging task due to the high risk decisions in the diagnosis, monitoring and care of patients. However, in rehabilitation of patients with limited or restricted mobility, as is the case of geriatric patients, IS have been used to control the positioning of robotic hospital beds to prevent the appearance of bed-pressure ulcers, also in activity monitoring and bed-rails

control [13,14,15,16]. Most of the systems for automatic control of hospital bed positions are based in the detection of the posture of patients on the bed using presence sensors, digital cameras, thermal cameras and mattress pressure sensors [19,20,21]. Modern hospital beds are able to adopt twelve positions ranging from the *home position* to the *sit-to-stand position*, but do not have a mechanism to prevent accidents when the change of positions takes place. Some of the most common accidents when the bed is moving are the down-fall of patients, bad posture when the bed is moving, and injuries caused by improper use of motion controls. In this research we propose an IS in order to reduce the risks of operating a hospital bed with multiple-positions. The main objective of our IS is to prevent accidents when the bed is moving, this is done by detecting the posture of patients using a mattress pressure sensor. We identify the posture of patients by performing an analysis and classification of the pressure distributions using an initial training set of correct postures for all bed positions. The proposed IS can be used to assists the medical staff and also provides a mechanism for an automatic and non-assisted control of bed positioning. Monitoring of leaning people is often used in a variety of hospital

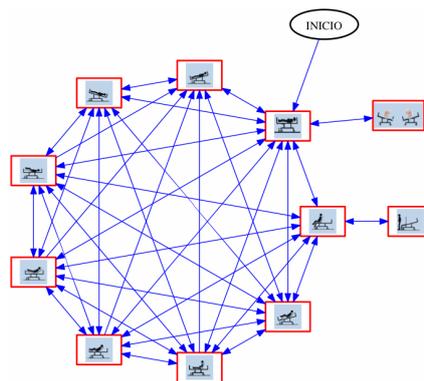


Fig. 1. Transition graph of all possible positions of the robotic hospital bed.

process as geriatrics, rehabilitation, orthopedics and even used in psychology and sleep studies [22]. Although patients monitoring is usually doctors and nurses activity, this can be automated to have better control of the patients in any time [20] doing human intervention only to rate different positions for which patient has passed, the time spent in each posture and incurred transitions. This automatization process is not new, as there are many attempts to recognize the position of a body in a given surface [20,13,14,16,15], most of the systems for posture recognition are based on the detection of the posture of patients on the bed, using presence sensors, digital cameras, thermal cameras or mattress pressure sensors [19,20,21]. Digital and thermal cameras achieve high resolution. However, they are affected by environment illumination, temperature

conditions and occlusions. Mattress pressure sensors are invariant to illumination and temperature conditions, all the same these sensors have a limited resolution and accuracy. Commercial pressure sensors are available with resolutions from 4 to 1024 sensor units, higher resolution are prohibitively expensive for many applications. Recent developments in the fabrication of low cost pressure matrix sensors are based on organic devices. In [23] a flexible 32×32 array of pressure sensors have been developed for skin sensitivity with a density of 10 units per inch, it is based on organic field-effect transistors. However, it is limited to small areas and are mostly used in applications such as tactile displays and mobile devices. In [24] pressure sensors based on organic thin film transistors was developed and applied to wearable electronics, textiles and skin for robots. On the other hand, proposed pressure sensors using resistive devices have been tested at highly sensitivity and very short response times that can be inexpensively fabricated over large areas [25,26]. Thus, given the high cost in the fabrication of array pressure sensors, is necessary a methodology that use the few data to solve the classification issue. We choose four basic positions according to the famous study by professor Chris Idzikowski [22], who classified the most common sleeping positions in six classes, that is, *foetus*, *log*, *yearner*, *soldier*, *freefaller* and *starfish*. These positions can be grouped in four basic classes: *left lateral decubitus* position or foetus, *right lateral decubitus* position of which the log and yearner positions belong to this class, *supine* position of which the soldier and starfish belong to this class, and *prone* position or freefaller. We use a flexible array pressure sensor with 448 units distributed in an area of 1860×886 mm. The pressure sensors are based on variable resistive sensors at 4096 pressure levels and normal resistance from 1 to 50 kiloOhm [27]. We construct a database with pressure levels (scaled to 0-255 range) transformed by a normalization process and apply it the HOG algorithm [17], the database contain samples of four postures showed in Fig. 1. With this database, we can construct a classification model using a *support vector machine* (SVM) classifier [28]. Section 2 explains the intelligent monitor which controls the robotic bed, in section 3 we refer to the image descriptors. Section 4 describes the way in which we make the recognition of the position of patients. Section 5 shows some results and finally Section 6 refers about conclusions.

2 Intelligence System for Automatic Control of Bed Positions

A robotic hospital bed is able to adopt several positions depending on the needs for a particular patient, also can be programmed to perform a series of movements in a period of time. We have a transition diagram (Fig. 1) of our robotic hospital bed of the most used positions by medical specialists. The initial position is the *home position* to which all other positions can reach, with the exception of the sit-to-stand position. The transitions of the robotic bed are performed by means of mechanical actuators, and the time it takes to go from one position to another depends on the weight of the patient and the current

position. When a transition is performed may happen that the patient falls or can be hurt by being in a bad posture, even when a specialist is operating the robotic bed. The IS will be able to detect if the patient is in a correct position in order to perform the requested transition, and will send a visible alert to prevent possible downfalls. The Fig. 3 shows the main stages of the IS for the posture recognition: in the initial stage the pressure distributions is obtained from the pressure sensor array, in the second and third stages an analysis and pre-processing is performed, also a feature extraction using HOG [17] and SHIFT [18] descriptors are applied over the pressure distributions that are considered as gray scale images; in the fourth stage a database of features is constructed, and in the last two stages we construct a model for feature classification and prediction, we compare the results of three classifiers such as *Support Vector Machines*, *Decision Trees* and *Bayes-naives Networks*. To simplify the posture recognition we consider three basic postures: the *right lateral decubitus*, *supine* and the *left lateral decubitus* positions (see Fig. 2), and since that the *prone* position is almost the same as the supine position, its detection is achieved by an analysis of the pressure distribution. The Fig. 2 shows the three basic correct positions displayed as gray scale images, obtained from simulated data of the pressure sensor array.

	
Right lateral decubitus (RLD)	Supine
	
Left lateral decubitus (LLD)	

Fig. 2. Three basic correct positions obtained from simulated data.

Thus, the database of features (see Fig. 3) is constructed such as it contains three set of equal size of the three basic positions and an appropriate number of its variants. The database of features is then used to construct the model of the classifier in order to make predictions of pressure distributions that do not belong to the database.

3 Image Descriptors

Finally, the IS for posture recognition can be used to control the actuators of the robotic hospital bed in a semi-or automatic way, and can prevent accidents when the bed is moving slowly. Thus, when a bad posture is detected then the

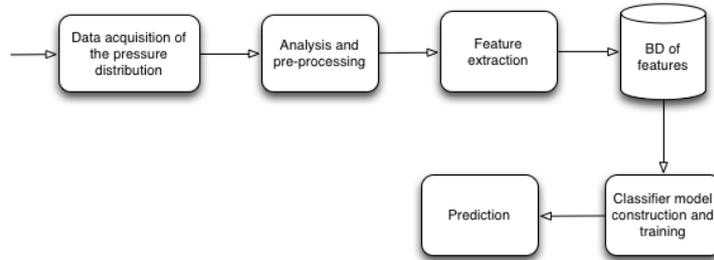


Fig. 3. Main blocks of the IS for posture recognition.

IS send a signal to the actuators either to stop the transition movement or to return to the previous position. A descriptor is a representation of an image, it characterizes the most important features of interest. Descriptors are used in many applications such as image classification and recognition. For these tasks, the desired properties of an image descriptor are the invariance with respect to rotation, scaling, perspective and illumination, several descriptors exist as SIFT [32], SURF [33], DENSE-SIFT [34] and others, but given the features of our image (low quality and gray scale) and mainly due their low computational cost, we decide to use the HOG descriptor, in experiments, HOG showed superior results at runtime compared to SIFT, Dense-SIFT and SURFT; commonly used in posture recognition works [35].

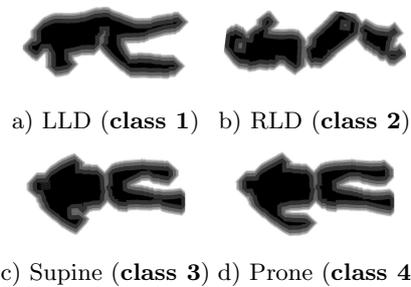


Fig. 4. Simulated basic posture positions.

3.1 HOG descriptor

HOG (*Histogram of Oriented Gradients*) descriptor is used in computer vision and image processing for object detection. This descriptor is based on the number of occurrences of the gradient directions in specific parts in an image. The main idea behind the HOG descriptor is that the local shape and appearance can

be described as the distribution of gradient intensities. Based on that fact, an image is divided in several overlapping blocks, and for each block a histogram of gradient directions is computed. Thus, the HOG descriptor is the composition of all histograms for each block. Fig. 5 shows an example of a simulated pressure image and its gradient directions. The size of the HOG descriptor depends on the size of the blocks and on the number of bins in which the interval from 0 to 180 degrees is divided. We use the HOG descriptor to characterize each pressure image. Therefore, for each pressure map, a vector of gradient directions is constructed. We consider that the histogram of gradient directions can be used for image classification, since that small changes in the pressure images are easily detected as changes in gradient directions.



Fig. 5. From left to right: simulated pressure map and image gradient directions.

4 Methodology for Posture Recognition

The proposed methodology takes the raw data of sensors and transforms it into HOG image, this representation as HOG image will be used as input in one SVM classifier, the final output is the classifier prediction about the position. Our system receives as input one array of 448 elements, with pressure levels (0-4096 units of pressure), each of which represents one sensor in the surface where a person is lying down, these sensors are distributed in 32 rows and 14 columns. Fig. 6 shows the main blocks of the proposed methodology for posture recognition. In the initial stage, the pressure distribution is obtained from the pressure sensor array, data obtained of the sensors is in range of 0 to 4096 levels of pressure, where 0 is the maximum pressure possible and 4096 is null pressure. Then, in the second block we transform raw data taking three considerations, first, the pressure applied by human body (considering weight between 40 to 150 Kg) is between 2500 to 4096 in the scale of pressures (see Fig. 7), then we can cut the range only to human body requirements and gain definition. Second consideration is an array escalation from 2500-4096 scale to 0-255 scale for process the array as gray scale image in the next block. The final consideration of the second block is applying a scale algorithm for images to obtain interpolated image of raw pressure data. In third stage, we use a feature extraction using HOG

descriptor and applying it to the pressure distributions that are considered as gray scale images. In the stage delimited with broken line, we make the SVM model, the first three stages are repeated several times with one human body in different positions (we consider positions described in Fig. 3), and we make a data base with these, after we use these data base for make a SVM model, how we know the position in the which was taken each measurement, we have a pressure interpolated and classified data base, then we can construct a SVM model that predicts the four positions of the Fig. 3. When SVM model is ready, we can use it in order to monitor person movements, the fourth stage implies to take this SVM model and to use it as input the first three stages output, then the fifth stage have an accurate prediction.

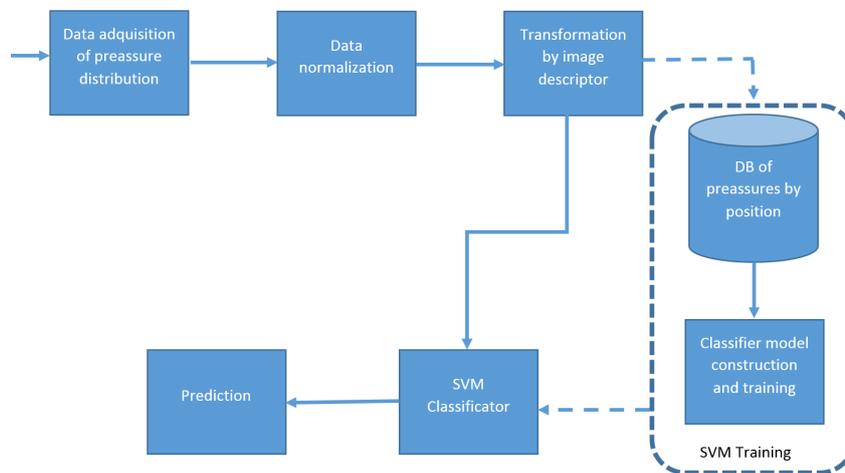


Fig. 6. Main blocks of methodology for posture recognition

5 Results

5.1 Pressure Maps Database

The pressure sensors array consists of two independently modules with a resolution of 16×14 units in an area of 930×886 mm. Each pressure sensor has a response between 0 to 25 kg/cm². We develop an interface to synchronize the two modules at a frequency of 100 measurements per second. We construct four different pressure map data set for classification: Each data set consists of four classes, for every posture position (see Fig. 3). For each class we generated 501 instances. We choose this number of instances according to study in [36]. Each data set has a number of 2004 instances and each data set in stored as individual data base, only after of applying the methodology described in section 4. Fig. 8

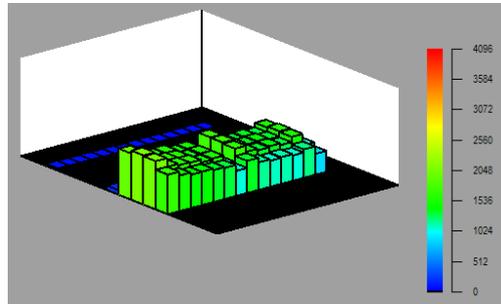


Fig. 7. Pressure levels obtained from posture of a person.

shows some examples of interpolated pressure maps. We normalize the range of values of each attribute vector, this was done in order to improve the performance of the classifier. As we see in Fig. 8 the pressure maps obtained are more intuitive than the showed in Fig. 7.

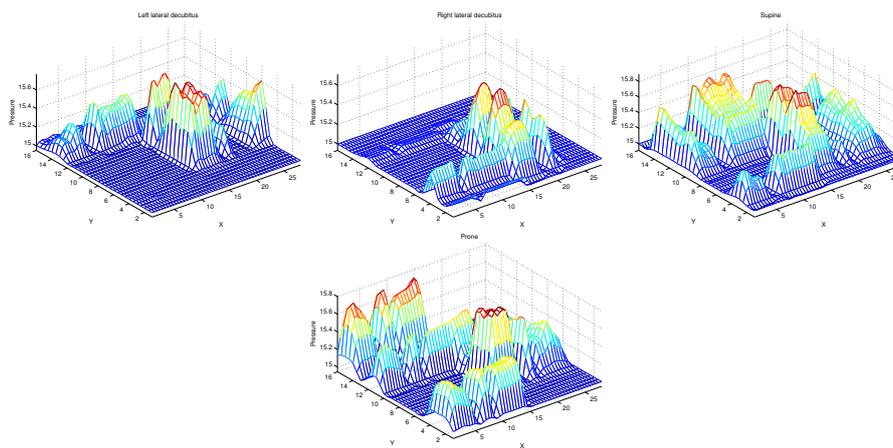


Fig. 8. Interpolated pressure maps corresponding to the four basic posture positions (see Fig. 3) before of apply the proposed methodology.

5.2 Posture Classification Using Pressure Maps

We use the LibLINEAR and LibSVM toolbox to perform the posture classification. Table 1 shows the results of the classification using the three databases with the proposed methodology. The classification results were obtained using 10-fold cross-validation. As can be seen the SVM with a linear kernel generates good results using the raw and interpolated data. Using image descriptors, however,

has a better performance with a RBF kernel. The advantage of using image descriptors is that the time to construct the model of the classifier is slower than using raw and interpolated data. We need not only the percentage of correctly classified instances, for a real-time process we need to verify the computing time, logically the data sets with many elements will take longer time, but we need to found the balance between assertiveness and runtime. Table 2 shows the time that take each dataset for be processed by cross-validation in one computer with processor Intel pentium T4500 (2.3GHz). Also, it shows results which are obtained by using cross-validation algorithm that process 35,000 sensor arrays inputs, then in this table we can show the number of sensor arrays what can be processed per second. Taking the results of tables 1 and 2, we can see

Tabla 1. Percentage of correctly classified instances.

	Linear-SVM	RBF-SVM
640×640	51.05 %	54.10 %
320×320	96.21 %	98.57 %
100×100	98.54 %	99.01 %
32×14	84.32 %	85.89 %

Tabla 2. Percentage of correctly classified instances.

	Time (seconds)	Arrays per second
640×640	5412	6
320×320	977	34
100×100	72	486
32×14	17	2058

that the better performance is obtained by 100×100 interpolated data set, that has 99.01% of assertiveness and can process 486 data arrays per second (although the hardware only give 100 per second). Using images of patients involving different positions, the intelligent system is trained so that it was able to recognize different postures of patients on a simulator of the bed that was built (Fig. 5.2). The intelligent monitor helps to know if the patient is on a correct position in each different position of the bed. There are different positions that patients may have on the bed. These positions will be processed for analysis in different positions that the patient may have on the bed while this robot could be moving. These positions will be processed for real-time analysis and from that time the intelligent system can perform posture recognition right or wrong in a way that could indicate a total stoppage of the movement of the robotic bed. Fig. 5.2 show part of results of the data processing applied to the images obtained from the movements of the bed. After this step, then the artificial intelligent



Fig. 9. Some positions of the robotic bed.

algorithm works in order to identify if the read position of patient is the right for the current position of the bed. Our team have developed a graphical interface

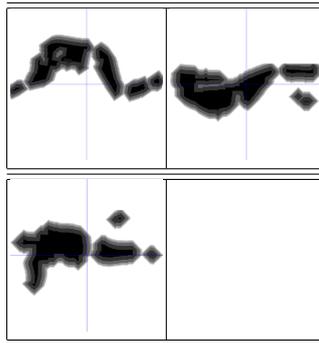


Fig. 10. Data pre-processing before the recognition step.

which help to visualize the results in terms of the identification and classification of the positions of the patient considering that robotic bed can be moving (Fig. 5.2) .

6 Conclusions

Our device has a system which represents a strategy to endow robots assistants ability to detect risk scenarios to patients. In this case, when the robotic bed is moving can generate situations of risk if patients perform any bodily movement that is inappropriate to such a move. This represents an advance in medical care since through such devices can treat patients without medical personnel risk of an injury when moving patients personally. In this case, the robotic bed can be programed in such way the appropriate medical personnel may attend

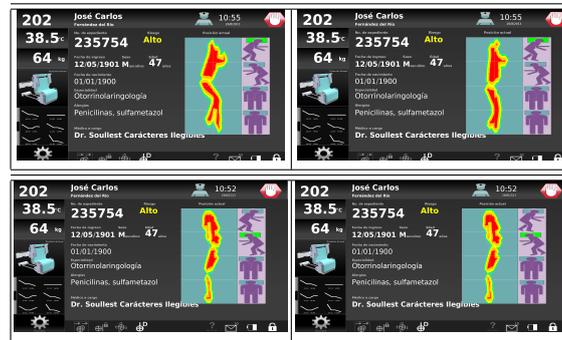


Fig. 11. A graphical interface. Patients positions recognized from patient movements data. The robotic bed could be moving.

to other patients while applying certain movement therapy automatically by medical monitoring. We have presented a methodology which is useful to classify the human bodies lying for real-time execution, the percentage of assertiveness is high, 99.01%, this ensuring correct monitoring in the majority of cases, also we show the correct size of interpolated arrays to get the best results, so that the system is able to process the 100 arrays per second that hardware is able to deliver. These results allow us to observe that it is possible to identify in real time what the positions of the patient on the bed are. This allows us then command the robotic bed so that the mechatronic device can meet the intelligent monitor commands indicates. We are talking about the intelligent system would act as a robot controller and receives information from the environment. Today, we are currently developing the sub system to help us identify trends movement in order the robot bed reacts before the patient execute some movement. This will help prevent multiple scenarios of risk to patients. Likewise, we are working to develop strategies to ensure 100% detection of the positions of the patient to avoid the risk that is involved in the remaining 1% so far can not meet our development. It is important to emphasize that this intelligent device was designed and manufactured considering specific requirements indicated by medical personnel of Mexican public hospitals as previously explained. This is very important because the whole system responds to a social need and not stems from of an offer by the research experience. Our development is in the process of intellectual property protection, including all aspects of intelligent medical device. We believe that at present, the development of technology must focus on the specific demands of the market and society requires and not all should concentrate on proposals that scientific and research experience to offer. In this way, we would be connecting the strength of technology development with actual needs around us.

References

1. Wang, Y., Butner, S. E., Darzi, A.: The Developing Market for Medical Robotics. *Proceedings of the IEEE*, Vol 94(9), pp. 1763–1771 (2006)
2. Ávila-Chaurand, R., Prado-León, L. R., González-Muoz, E. L.: Dimensiones antropométricas de población latinoamericana: México, Cuba, Colombia, Chile. Universidad de Guadalajara, Centro Universitario de Arte, Arquitectura y Diseño, División de Tecnología y Procesos, Departamento de Producción y Desarrollo, Centro de Investigaciones en Ergonomía, 207 p. (2001)
3. Pavlovic, N. D., Milosevic, M., Pavlovic, N. T.: Development of Mechanisms for Adjusting Positions of a Multifunctional Bed. In: 13th World Congress in Mechanism and Machine Science, Guanajuato, Mexico, pp 1–6 (2011)
4. Mohammeda, M. N., Khrita, N. G., Abdelgneia, M. A., Abubakera, Es., Muftaha, A.F., Omara, M. Z., Salleha, M. S.: A New Design of Multi-Functional Portable Patient Bed. *Jurnal Teknologi (Sciences & Engineering)*, Vol. 58, pp. 61–66 (2012)
5. Ching-Hua, W., Ting-Chun, T., Shin-Chieh, H., Wan-Chun, C., Yen-Ming, C., Kun-Tse, T., Chun-Went, Y., Kuo-Yil, C.: Hospital Bed with Auxiliary Functions of Lateral Positioning and Transferring for immobilized patients. In: The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Taipei, Taiwan, pp. 2991–2995 (2007)
6. Shih-Wei P., Feng-Li L., Li-Chen F.: Mechanism Design and Mechatronic Control of a Multifunctional Test Bed for Bedridden Healthcare. *IEEE/ASME Transactions on Mechatronics*, Vol. 15(2), pp. 234–241 (2010)
7. Kittipichai, R., Ariyarat, A.: The Sizing Optimization of Hospital Bed Structure for Independently Supporting Left and or Right Leg Using Genetic Algorithms. *International Journal of Modeling and Optimization*, Vol. 11 no. 2, pp. 122–128 (2011)
8. Kap-Ho, S., Changmok, O., Tae-Yong, C., Ju-Jang, L.: Bed-type Robotic System for the Bedridden. In: *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, California, USA, pp. 1170–1175 (2005)
9. Andhare, A., Onkar, A., Padole, P.: Design of Bed for Bedridden Patients: Analysis and Synthesis of Mechanisms. In: 15th National Conference on Machines and Mechanisms, Chennai, Tamilnadu, India, pp. 1–6 (2011)
10. Vázquez-Santacruz, E., Gamboa-Zúñiga, M.: A diagnosis methodology for assistive technology development. In: *Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2013 10th International Conference on, pp. 163–169 (2013)
11. Beltrán-Herrera, A. Vázquez-Santacruz, E., Gamboa-Zúñiga, M.: Real-Time Classification of Lying Bodies by HOG Descriptors, In: *Pattern Recognition - 6th Mexican Conference, MCPR 2014*, Cancun, Mexico, pp. 211–220 (2014)
12. Mehta, R. K., Horton, L. M., Agnew, M. J., Nussbaum, M. A.: Ergonomic evaluation of hospital bed design features during patient handling tasks. *International Journal of Industrial Ergonomics*, vol. 41, pp. 647–652 (2011)
13. Nicol, Klaus; Rusteberg, Dietmar: Pressure distribution on mattresses. *Journal of Biomechanics*, vol. 26, number 12, pp. 1479–1486 (1993)
14. James W. DeVocht, David G. Wilder, Eric R. Bandstra, Kevin F. Spratt: Biomechanical evaluation of four different mattresses. *Applied Ergonomics*, vol. 37, number 3, pp. 297–304 (2006)

15. Townsend, D., Holtzman, Megan, Goubran, R., Frize, M., Knoefel, F.: Relative Thresholding With Under-Mattress Pressure Sensors to Detect Central Apnea. Instrumentation and Measurement, IEEE Transactions on, vol. 60, number 10, pp. 281–3289 (2011)
16. Amaya Arcelus, Christophe L. Herry, Rafik A. Goubran, Frank Knoefel, Heidi Sveistrup, Martin Bilodeau: Determination of Sit-to-Stand Transfer Duration Using Bed and Floor Pressure Sequences. IEEE Trans. Biomed. Engineering, vol. 56, number 10, pp. 2485–2492 (2009)
17. Dalal, Navneet; Triggs, Bill: Histograms of Oriented Gradients for Human Detection. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Volume 01, CVPR '05, pp. 886–893, IEEE Computer Society (2005)
18. Lowe, David G.: Object Recognition from Local Scale-Invariant Features. In: Proceedings of the International Conference on Computer Vision, Volume 2, ICCV'99, IEEE Computer Society (1999)
19. Robert Grimm; Sebastian Bauer; Johann Sukkau; Joachim Hornegger; Günther Greiner: Markerless estimation of patient orientation, posture and pose using range and pressure imaging. Int. J. Computer Assisted Radiology and Surgery, Volume 7, number 6, pp. 921–929 (2012)
20. Manuel Chica; Pascual Campoy; María A. Pérez; Tomás Rodríguez; Rubén Rodríguez; Óscar Valdemoros: Corrigendum to: Real-time recognition of patient intentions from sequences of pressure maps using artificial neural networks. Computers in Biology and Medicine, volume 43, number 9, pp. 1302 (2013)
21. Kap-Ho Seo; Tae-Yong Choi; Changmok Oh: Development of a robotic system for the bed-ridden. Mechatronics, volume 21, number 1, pp. 227–238 (2011)
22. Idzikowski, Christopher: Learn to Sleep Well. Watkins (2010)
23. Someya, Takao; Sekitani, Tsuyoshi; Iba, Shingo; Kato, Yusaku; Kawaguchi, Hiroshi; Sakurai, Takayasu: A large-area, flexible pressure sensor matrix with organic field-effect transistors for artificial skin applications. In: Proceedings of the National Academy of Sciences of the United States of America, volume 101, number 27, pp. 9966–9970 (2004)
24. Manunza I., Bonfiglio, A.: Pressure sensing using a completely flexible organic transistor. Biosensors and Bioelectronics, volume 22, number 12, pp. 2775–2779 (2007)
25. Mannsfeld, S. C. B., et. al.: Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers. Nature Materials, volume 9, number 10, pp. 859–864 (2010)
26. Sekitani T., Zschieschang, U., Klauk, H., Someya, T.: Flexible organic transistors and circuits with extreme bending stability. Nature Materials, volume 9, number 12, pp. 1015–1022 (2010)
27. Sensing Tex Smart Textiles, Webpage: <http://www.sensingtex.com/> (2013)
28. Byun, H., Lee, S.-W.: A survey on pattern recognition applications of support vector machines. International Journal of Pattern Recognition and Artificial Intelligence, volume 17, number 03, pp. 459–486 (2003)
29. Keerthi, S. Sathiy; Lin, Chih-Jen: Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel. Neural Comput., July, volume 15, number 7, pp. 1667–1689, MIT Press (2003)
30. Cortes, Corinna and Vapnik, Vladimir: Support-Vector Networks. Mach. Learn., volume 20, number 3, pp. 273–297, Kluwer Academic Publishers (1995)
31. Ting, KaiMing: Confusion Matrix. In: Encyclopedia of Machine Learning, Sammut, Claude and Webb, Geoffrey (eds.), Springer US, pp. 209–209 (2010)

32. Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. Kluwer Academic Publishers, pp. 91–110, volume 60, number 2 (2004)
33. Bay, Herbert; Ess, Andreas; Tuytelaars, Tinne; Van Gool, Luc.: Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, volume 110, number 3, pp. 346–359, Elsevier Science Inc. (2008)
34. Wang, Jian-Gang; Li, Jun; Lee, Chong Yee; Yau, Wei-Yun: Dense SIFT and Gabor descriptors-based face representation with applications to gender recognition. In: *ICARCV*, pp. 1860–1864, IEEE (2010)
35. Dalal, N. and Triggs, B.: Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition, CVPR 2005*, IEEE Computer Society Conference on, volume 1, pp. 886–893 (2005)
36. , Hsu, C.-W., Chang, C.-C., and Lin, C.-J.: A practical guide to support vector classification. Tech. rep., Department of Computer Science, National Taiwan University (2003)

Captura de datos para análisis de la dinámica del tecleo de números para sistema operativo Android

Selene Nieto-Ruiz, Yonic A. Gómez-Sánchez, Asdrúbal López-Chau, Carlos A. Rojas

Ingeniería en Computación, Centro Universitario UAEM Zumpango, Zumpango, Estado de México, México

Resumen. El análisis de la dinámica del tecleo es una técnica usada para verificar la identidad de usuarios de sistemas informáticos. Actualmente, existen pocos conjuntos de datos públicamente disponibles de usuarios reales, y los que se encuentran corresponden a teclados tipo QUERTY. El teclado tipo numérico, que es usado en diversas aplicaciones importantes, ha recibido poca atención por parte de la comunidad científica. En este trabajo se presenta el desarrollo de un sistema para sistema operativo Android, cuya finalidad es la captura de los tiempos de tecleo usados comúnmente en biometría informática. Los datos generados por usuarios reales usando este sistema, se han publicado en la Internet para su libre descarga. En este artículo se presenta un resumen de los datos y se presentan trabajos futuros.

Palabras clave: dinámica de tecleo, biometría, Android.

1. Introducción

La biometría informática se refiere al conjunto de técnicas, principalmente estadísticas, que son aplicadas para verificar la autenticidad de individuos, tomando como base sus rasgos físicos o de conducta. La biometría informática se divide en estática y dinámica. La primera usa las características físicas de los sujetos, mientras que la segunda emplea sus rasgos conductuales. Entre los rasgos físicos más utilizados se encuentran las huellas dactilares y algunas partes del ojo como la retina o iris. Algunos ejemplos de características conductuales incluyen la voz, la forma de firmar, y el modo de caminar.

Recientemente, el grupo de hackers " *Chaos Computer Club*"¹ demostró que utilizar el tipo de biometría informática estática para proteger sistemas ya no es confiable. Ya que mediante simples fotografías tomadas con teléfonos inteligentes es posible copiar las huellas dactilares. Debido a lo anterior, la biometría informática dinámica cobra mayor importancia en diversas aplicaciones, y es actualmente usada cada vez más en sitios Web ² y sistemas informáticos³.

¹ <http://www.ccc.de/en/>

² Por ejemplo, <https://mega.co.nz>

³ Ejemplo, Sistemas de Administración Tributaria en México

La dinámica del tecleo (*keystroke dynamics*) [13], que consiste en usar la forma o el ritmo en que un usuario escribe en un teclado para autenticar su identidad, es una de las principales técnicas de la biometría informática dinámica. Numerosos trabajos y proyectos han sido publicados recientemente [4], [13], ya sea proponiendo nuevos algoritmos [7], implementando o aplicando la dinámica del tecleo en diferentes dispositivos. Generalmente, los estudios con dinámica de tecleo se han centrado en teclados tipo QUERTY [14]. Sin embargo, son pocos los trabajos que se enfocan en teclados tipo numérico. En la actualidad, son varias las aplicaciones en las que se utiliza este tipo de teclado, por ejemplo, cajeros automáticos, cajas fuertes, teléfonos [3] y para captura en hojas de cálculo.

En este artículo, se presenta la implementación completa de un sistema para la captura y almacenamiento de la dinámica del tecleo en dispositivos con sistema operativo (SO) Android, para teclados tipo numérico. El sistema propuesto, fue implementado y probado con 14 usuarios reales.

2. Biometría informática

El término biometría, proveniente del griego "*bios*" vida y "*metron*" medida, se refiere a sistemas de reconocimiento humano a través de la medición de sus rasgos físicos y/o de conducta [5]. El uso de técnicas para la identificación de individuos se remonta a tiempos antiguos. La medición de algunas partes del cuerpo fue usada primero, y luego la impresión de huellas dactilares [10]. A principios de la década 2000, otras características físicas que empezaron a utilizarse fueron el iris del ojo y los patrones vasculares de la retina, ya que se descubrió que son únicos para cada ser humano [12].

Existe dos modos de emplear la biometría, para identificación o para verificación. En el primero, conocido como igualación 1 a N, un sistema identifica a un individuo de entre una población que se encuentra registrada. Para ello se realiza una búsqueda de coincidencia, basándose en algún rasgo particular. En el modo de verificación, se comprueba la identidad de un individuo, usando un patrón que ha sido previamente registrado del mismo; esto también es llamado igualación 1 a 1.

La biometría se clasifica en dos grandes tipos [9]:

Biometría Estática

Se caracteriza por considerar parámetros derivados de la medición directa de alguna característica fisiológica compleja de cuerpo humano. El adjetivo estática es debido a que, bajo condiciones normales, los valores de la característica medida no varían significativamente durante el ciclo de vida de un ser humano. Las principales aplicaciones y estudios se basan en los sistemas biométricos de huellas dactilares, geometría de la mano, iris, entre otras.

Biometría Dinámica

Los psicólogos han mostrado que los seres humanos somos predecibles en nuestros comportamientos al desempeñar tareas repetitivas o rutinarias. La biometría dinámica es la encargada de estudiar las características o rasgos de la conducta para verificar la identidad de las personas, a diferencia de la biometría

estática en la dinámica los rasgos o características varían en una misma persona a lo largo del tiempo.

En la siguiente sub-sección, se presenta una de las técnicas de biometría dinámica que se ha empleado cada vez más en aplicaciones recientes, y que se estudia en este artículo.

2.1. Dinámica del tecleo

Actualmente, es común para la mayoría de las personas contar con una computadora, teléfono inteligente o tableta electrónica. Una de las principales actividades que se realizan en estos dispositivos, es la escritura por medio del teclado tipo QWERTY, que es el predominante en este tipo de sistemas.

Otro tipo de teclado que es empleado en algunas aplicaciones, es el numérico, cuya apariencia es presentada en la Figura 1. Ejemplo de estas aplicaciones son calculadoras simples, pantallas para ingreso al sistema y pantallas para captura de números, como en la banca electrónica.

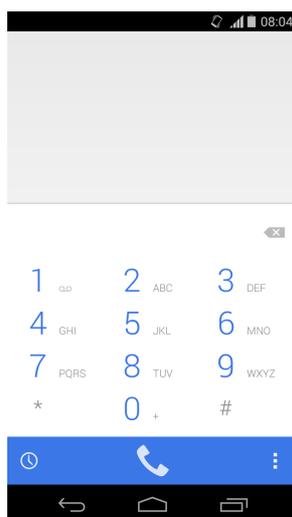


Fig. 1: Teclado tipo numérico de Android

- Tiempo de presión (*dwell-time*): Es el intervalo que transcurre cuando el usuario mantiene presionada y libera la misma tecla.
- Tiempo de cambio (*flight-time*): Se considera el tiempo que transcurre cuando el usuario suelta una tecla y presiona la tecla siguiente.

A diferencia de las características físicas de una persona, que son inmutables a lo largo del tiempo (bajo condiciones normales, es decir, sin accidentes,

enfermedades deformadoras, etc.), la forma de teclear de una persona puede sufrir variaciones notables a lo largo de su vida. Existen distintas razones por lo que sucede esto, por ejemplo, el grado de atención al teclear, o estado de salud de la persona. Las características físicas del teclado utilizado (tamaño, separación de teclas, etc.) tienen también una influencia notable sobre el valor de *dwell-time* y *flight-time*.

3. Sistema Operativo Android

Android es un sistema operativo (S.O.) orientado a dispositivos móviles, y está basado en Linux [6]. Fue desarrollado en 2003, en Palo Alto, California, por Andy Rubin, Rich Miner, Nick Sears y Chris White. Desde 2005, Android es propiedad de Google.

El código fuente de Android sobrepasa los 10 millones de líneas, escritas en varios lenguajes de programación como C, C++ y Java. Este sistema operativo está compuesto por los siguientes componentes: aplicaciones, marco de aplicaciones, bibliotecas, máquina virtual de Java reducida (llamada Dalvik) y núcleo del sistema.

Al crear un proyecto nuevo en el entorno Android, automáticamente se generan los siguientes directorios [8]:

- src: Contiene en su totalidad el código fuente (Java) de la aplicación desarrollada.
- res: En este directorio se encuentran los recursos necesarios para generar la aplicación.
 - res/drawable: Almacena imágenes en diversos subdirectorios dependiendo de su resolución.
 - res/raw: Contiene los archivos que son de propósito general, excepto los formatos XML.
 - res/layout: Se encuentran los archivos que definen la interfaz gráfica, generalmente siempre en XML.
 - res/values: Guarda datos, tales como colores, cadenas de texto, estilos.
- gen: Se almacena un conjunto de archivos al momento de que el proyecto se compila, con el fin de dirigir los recursos de la aplicación.
- assets: Se encuentran los archivos indispensables para el correcto funcionamiento de la aplicación, por ejemplo, archivos de datos o de configuración.
- Archivo AndroidManifest.xml: Se considera el archivo más importante para la creación de una aplicación. Este se genera automáticamente y se encuentra definida la configuración del proyecto XML.

La creación de interfaces gráficas puede realizarse utilizando únicamente código, o combinando código fuente y XML.

4. Trabajos relacionados

Son variados los trabajos que tratan sobre verificación de la autenticidad de usuarios basada en la dinámica del tecleo o uso del dispositivo apuntador ratón.

De acuerdo a el artículo [1], se presenta una nueva forma de sistema biométrico, basado en la dinámica del mouse o ratón. Se desarrolla una técnica para modelar características del comportamiento de los datos capturados con ayuda de redes neuronales artificiales. Esta investigación se centra en la extracción de características de comportamiento que son relacionadas con el usuario. Para ello se utiliza técnicas de estadísticas como lo es redes neuronales. Se muestra que se logra un FAR (por sus siglas en inglés False Acceptance Rate) de 2.4946% y un False Rejection Rate (FRR) de 2.4624%.

El FAR y FRR son métricas para determinar el desempeño de un sistema de detección de intrusiones.

- FAR: mide la probabilidad de que un impostor puede ser erróneamente aceptado por el sistema.
- FRR: mide la probabilidad de que un usuario genuino pueda ser rechazado por el sistema.

Para su realización se les pidió a participantes jugar un juego donde tenía que hacer clic lo más rápido posible en un cuadro en movimiento durante un periodo de tiempo fijo. Lo cual permitía recoger las coordenadas del ratón y calcular varias características tales como velocidad, aceleración, velocidad angular, curvatura, entre otros más. Para la autenticación del individuo se hace mediante la comparación de dicha información contra algunos umbrales.

En el artículo [11] se presenta un modelo para la autenticación de usuarios en función de pulsaciones de teclado en dispositivos móviles.

Este modelo captura el tiempo el tiempo necesario para que el usuario presione y suelte las teclas cuando este escribiendo, para que de esta manera se compare la información con el registro almacenado anteriormente. Entonces, si el resultado es el mismo al registrado, el usuario se considera autentico e idóneo para acceder al sistema de lo contrario es considerado como un impostor por consecuencia el sistema queda bloqueado. Con ayuda del método de distancia Euclidiana alcanzo un FAR de 12.97% y un FRR de 2.25%.

Cuando se introduce la contraseña el usuario ejecuta su propia característica a través del acto de empuje y soltar. Lo que permite la identificación y autenticación mediante su ritmo de escritura (velocidad, tiempo de pulsación de cada tecla y liberación de cada uno).

La realización de este tipo de sistemas permite el aumento de seguridad móvil con el mejoramiento de control de acceso.

5. Principal aportación

La principal aportación de este proyecto es generar un conjunto de datos con tiempos de presión y tiempos de cambio de usuarios reales, lo anterior con la finalidad de que estos datos sean usados en estudios de biometría dinámica del tecleo. A diferencia de otros conjuntos de datos similares, en los que se emplean teclados tipo QUERTY para los experimentos, en nuestra propuesta los tiempos registrados corresponden a teclado tipo numérico. De acuerdo a la investigación documental realizada previamente a la realización de este trabajo, no existe actualmente un conjunto de datos públicamente disponible como el que se presenta en este artículo.

6. Sistema desarrollado para la captura de datos de dinámica del tecleo

El teclado tipo QUERTY es uno de los más utilizados en dispositivos móviles actualmente. Sin embargo, todavía son varias las aplicaciones importantes que utilizan teclados de tipo numérico. Algunas de ellas son las cajas fuertes, teclados en cajeros automáticos y otros sistemas de control de acceso. Pocos trabajos han prestado atención a la biometría dinámica con este tipo de teclado. En este trabajo, se desarrolla una aplicación para SO Android ⁴, que permite capturar los tiempos de presión y de cambio. Estos son almacenados en una base de datos local, para su posterior análisis.

La interfaz gráfica de usuario (GUI) que se diseñó, es similar a las de acceso a sistemas informáticos, y se muestra en la Figura 2.



Fig. 2: Interfaz para captura de usuario y contraseña

⁴ <http://www.alchau.com/research/2015/keystrokeDinamycsAndroid.rar>

Esta GUI tiene controles para que los usuarios ingresen un nombre de usuario y una contraseña. Para preservar el anonimato de los usuarios y al mismo tiempo para distinguir entre los datos generados por cada uno, se usaron números como identificadores para los usuarios. De esta forma, el primer usuario tiene como identificador el valor 1, el segundo el 2, y así sucesivamente.

Con la finalidad de que los datos generados por cada usuario puedan ser comparables con los otros, se decidió que el nombre del usuario y la contraseña fueran iguales en todos los casos. El nombre de usuario es **94255701**, mientras que la contraseña es **416850293**.

6.1. Arquitectura del sistema

El diagrama de clases del sistema implementado es mostrado en la Figura 3. La clase *Gestor de usuarios* controla el acceso y consultas a la base de datos local. Cada usuario es identificado con un número entero. La clase *Usuario* mantiene los datos de cada usuario, como el id y contraseña. Una de las clases más importantes es la denominada *EscucharTeclado*. Esta clase contiene los métodos para calcular los tiempos de presión y de cambio. La API (*Application Programming Interface*) de Android ofrece una serie de clases para facilitar la captura de eventos del teclado. Una combinación de los eventos *onKeyDown()* y *onKeyUp()* [2] fue usada para capturar los tiempos requeridos.

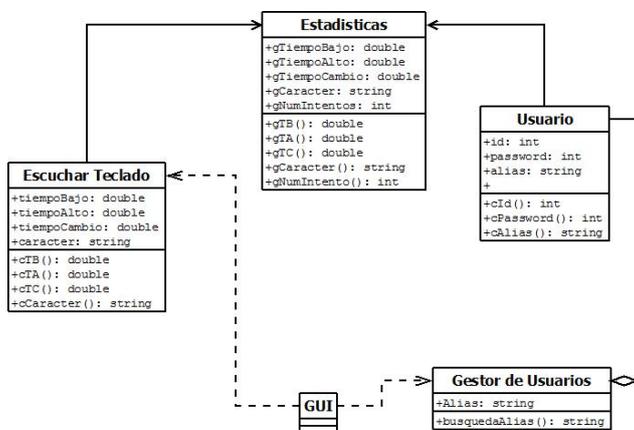


Fig. 3: Diagrama de clases del sistema implementado

7. Datos generados

El dispositivo en el que se probó el sistema, fue un teléfono inteligente modelo XT1058, con procesador ARMv7 (v71), pantalla de 4.7 pulgadas y resolución de

720 × 1184 p. El sistema operativo es Android 4.4.4 .

Se solicitó a 14 personas, que introdujeran el nombre de usuario (94255701) y la contraseña (416850293) un total de 10 veces. Para los casos en los que un usuario se equivoca, el sistema elimina ese registro y solicita que se intente nuevamente. Cada intento exitoso fue almacenado en una base de datos local en el teléfono inteligente.

Un resumen de los tiempos de presión promedio al introducir es el nombre de usuario y la contraseña es mostrado en las Figuras 4 y 5, respectivamente. Los tiempos de cambio promedio, son presentados en las Figuras 6 y 7.

Los datos correspondientes a todos los intentos de los 14 usuarios, han sido publicados en el sitio <http://www.alchau.com/research/2015/keystrokeData-1/data.zip>, y se encuentran disponibles para su descarga.

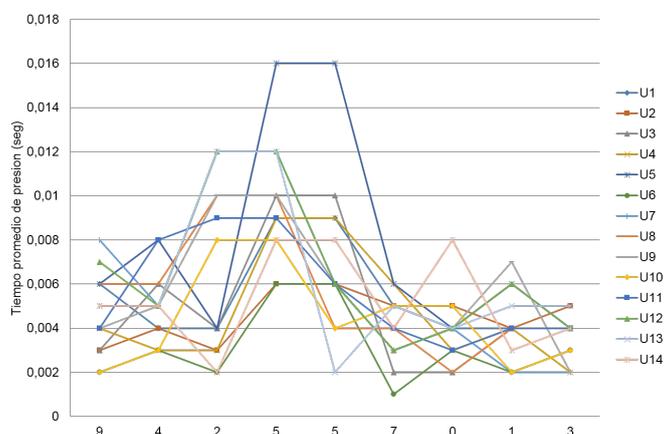


Fig. 4: Tiempos de presión promedio al introducir el nombre de usuario

8. Conclusiones y trabajo futuro

El análisis de la dinámica del tecleo ha sido estudiado ampliamente en la última década. En la literatura se reportan avances significativos sobre identificación de usuarios usando teclados tipo QWERTY. Sin embargo, de acuerdo a la investigación realizada en la presente investigación, no se ha prestado gran atención a estudios similares con teclados de tipo numérico.

En este artículo, se desarrolló una aplicación para sistema operativo Android, que permite capturar los tiempos de presión y de cambio en este tipo de teclado. Se realizaron pruebas con 14 usuarios reales, a quienes se solicitó capturar un mismo nombre de usuario y contraseña un total de 10 veces. Los datos generados para cada intento exitoso se han puesto disponibles públicamente en la Internet, para ser utilizados en futuros experimentos.

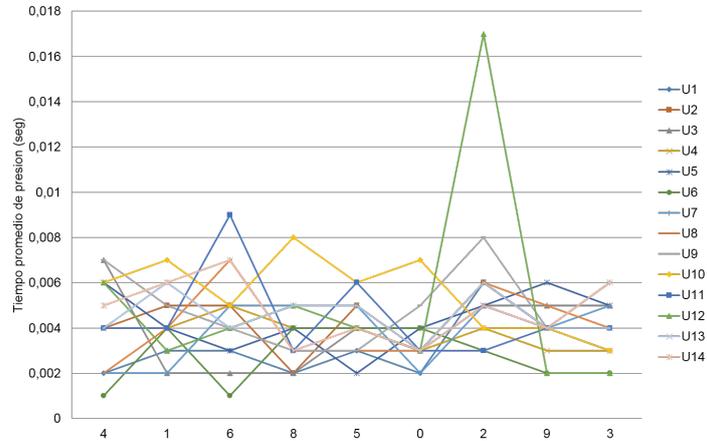


Fig. 5: Tiempos de presión promedio al introducir la contraseña

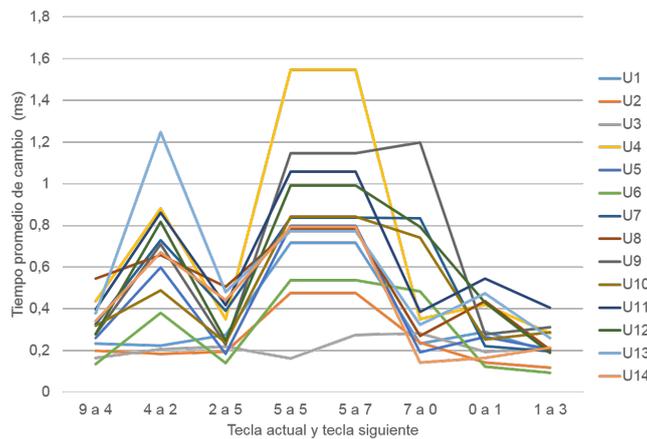


Fig. 6: Tiempos de cambio promedio al introducir el nombre de usuario

Como trabajo futuro, se tiene planeado probar algoritmos para verificar la identidad de usuarios basados en la dinámica de su tecleo. También se tiene considerado implementar este tipo de sistemas en dispositivos electrónicos, para ser utilizado con cerraduras electromecánicas. Por último, pero no menos importante, se tiene la intención de desarrollar un algoritmo propio para identificación de impostores.

Agradecimientos. Los autores de este artículo agradecen a los revisores anónimos por sus valiosas observaciones para mejorar este trabajo. También agradecen a la Universidad Autónoma del Estado de México por su apoyo económico, a través del proyecto de investigación 3790/2014/CID.

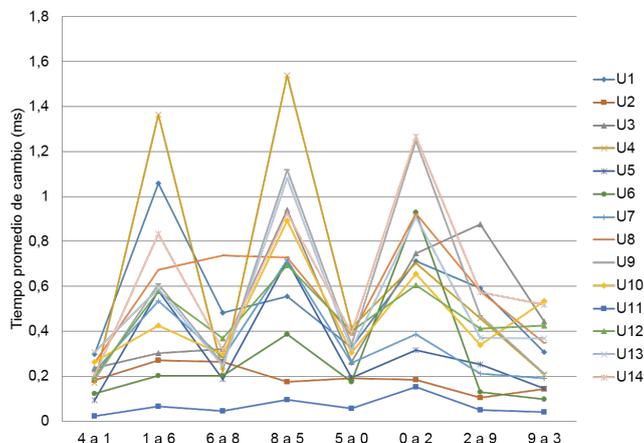


Fig. 7: Tiempos de cambio promedio al introducir la contraseña

Referencias

1. Ahmed, A.A.E., Traore, I.: A new biometric technology based on mouse dynamics. *IEEE Transactions on Dependable and Secure Computing* (2007)
2. Android: View.onkeylistener (Diciembre 2014), <http://developer.android.com/reference/android/view/View.OnKeyListener.html>
3. Antal, M., Szabó, L.Z., László, I.: Keystroke dynamics on android platform. *ELSEVIER* (2014)
4. Gaines, R., Lisowski, W., Press, S., Shapiro, N.: Authentication by keystroke timing: some preliminary results. *Rand Corporation* (1980)
5. Galván, G.I.: Sistema de autenticación de dispositivos móviles basado en biometría de comportamiento de tecleo (Junio 2007), <http://delta.cs.cinvestav.mx/~francisco/tesisIglesias.pdf>
6. Gironés, J.T.: *El gran libro de Android*. Barcelona (2012)
7. Obaidat, M.S., Macchiarolo, D.T.: An on-line neural network system for computer access security. *IEEE Transactions on Industrial Electronics* (1993)
8. Soriano, J.E.A.: *Android: Programación de dispositivos móviles a través de ejemplos*. Barcelona (2012)
9. UNAM: Clasificación de los sistemas biométricos (Octubre 2014), <http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/clasificaciontipo.html>
10. UNAM: Fundamentos de biometría (Noviembre 2014), <http://redyseguridad.fi-p.unam.mx/proyectos/biometria/fundamentos/antecedentes.html>
11. Urtiga, E., Moreno, E.: Keystroke - based biometric authentication in mobile devices. *IEEE Latin America Transactions* (2011)
12. Yampolskiy, R.V., Govindaraju, V.: Behavioural biometric: a survey and classification. *Int. J. Biometrics* (2008)
13. Yu, E., Cho, S.: Keystroke dynamics identity verification - its problems and practical solutions. *ELSEVIER* (2004)
14. Zhong, Y., Deng, Y., Jain, A.K.: Keystroke dynamics for user authentication

Reviewing Committee

Ricardo Acevedo	Ivan Adrian Lopez Sanchez
Moisés Alencastre Miranda	Antonio Marin-Hernandez
Roberto Alonso Rodriguez	Lourdes Martínez
Joanna Alvarado-Uribe	Martinez Medina
Gustavo Arroyo Figueroa	Miguel Angel Medina Perez
Christian Arzate	Efrén Mezura-Montes
Ivonne Maricela Ávila Mora	Sabino Miranda-Jiménez
Jorge Bautista López	Daniela Moctezuma
Ivo Buzon	Raul Monroy
Maria Victoria Carreras	Jaime Mora-Vargas
Felix Castro Espinoza	Saturnino Job Morales Escobar
Noé Alejandro Castro Sánchez	Lourdes Muñoz Gómez
Bárbara Cervantes	Antonio Neme
Jair Cervantes	Alberto Oliart Ros
Efren Chavez Ochoa	Mauricio Osorio
Gustavo Delgado Reyes	Elvia Palacios
Sofía N. Galicia-Haro	Hiram Eredín Ponce Espinosa
Natalia Garcia	Carlos Pérez Leguizamo
Alexander Gelbukh	Maricela Quintana López
David Gonzalez	Carlos A. Reyes-García
Miguel Gonzalez-Mendoza	Carlos Alberto Rojas Hernández
Hugo Gustavo González	Rafael Rojas Hernández
Hernández Mario Graff	Dafne Rosso
Fernando Gudiño	Oscar S. Siordia
Pedro Guevara López	Grigori Sidorov
Yasmin Hernandez	Abraham Sánchez López
Neil Hernandez Gress	Israel Tabarez Paz
Oscar Herrera	Eric Sadit Tellez
Rodolfo Ibarra	Nestor Velasco-Bermeo
Yulia Ledeneva	Elio Villaseñor
Asdrubal Lopez Chau	Alisa Zhila
Juan Carlos Lopez Pimentel	

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
mayo de 2015
Printing 500 / Edición 500 ejemplares

