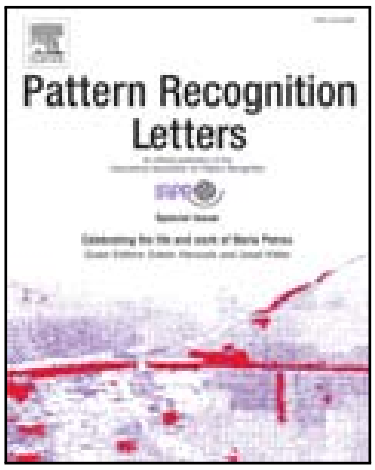

Pattern Classification using Smallest Normalized Difference Associative Memory

Rogelio Ramírez-Rubio, Mario Aldape-Pérez,
Cornelio Yáñez-Márquez, Itzamá López-Yáñez,
Oscar Camacho-Nieto

PII: S0167-8655(17)30050-8
DOI: 10.1016/j.patrec.2017.02.013
Reference: PATREC 6751

To appear in: *Pattern Recognition Letters*

Received date: 16 April 2016
Revised date: 13 February 2017
Accepted date: 14 February 2017

Please cite this article as: Rogelio Ramírez-Rubio, Mario Aldape-Pérez, Cornelio Yáñez-Márquez, Itzamá López-Yáñez, Oscar Camacho-Nieto, Pattern Classification using Smallest Normalized Difference Associative Memory, *Pattern Recognition Letters* (2017), doi: 10.1016/j.patrec.2017.02.013

Research Highlights (Required)

To create your highlights, please type the highlights against each `\item` command.

It should be short collection of bullet points that convey the core findings of the article. It should include 3 to 5 bullet points (maximum 85 characters, including spaces, per bullet point.)

- An Alpha-Beta associative model extension to real values is proposed
- A new associative classification algorithm is proposed
- Eight pattern classification problems from the medical field are addressed
- Twenty-one well known algorithms have been used to compare with our proposal
- Our proposal achieved the best classification accuracy averaged over all datasets

ELSEVIER

Pattern Recognition Letters
journal homepage: www.elsevier.com

Pattern Classification using Smallest Normalized Difference Associative Memory

Rogelio Ramírez-Rubio[a], Mario Aldape-Pérez[a,**], Cornelio Yáñez-Márquez[b], Itzamá López-Yáñez[a,b], Oscar Camacho-Nieto[a]

[a]*Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo IPN-CIDETEC, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, CIDETEC Building, Mexico City 07738, Mexico*
[b]*Instituto Politécnico Nacional, Centro de Investigación en Computación IPN-CIC, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, CIC Building, Mexico City 07738, Mexico*

ABSTRACT

In this paper a new associative classification algorithm is presented. The proposed algorithm overcomes the limitations of the original Alpha-Beta associative memory, while maintaining the fundamental set recalling capacity. This algorithm has two phases. The first phase is based on an Alpha-Beta auto-associative memory, which works in the domain of real numbers, unlike the traditional Alpha–Beta associative memories. In the second phase, normalized difference between the results of first phase and every pattern of the fundamental set is calculated. In order to demonstrate the behaviour and accuracy of the algorithm, multiple well known datasets and classification algorithms have been used. Experimental results have shown that our proposal achieved the best performance in three of the eight pattern classification problems in the medical field, using Stratified 10 Fold cross-validation. Our proposal achieved the best classification accuracy averaged over the all datasets addressed in the present work. Experimental results and statistical significance tests, allow us to affirm that the proposed model is an efficient alternative to perform pattern classification tasks.

1. Introduction

Early models of learning matrices appeared more than four decades ago (Steinbuch, 1961; Steinbuch and Frank, 1961; Steinbuch, 1964), and since then associative memories have attracted the attention of major research groups worldwide (Verbiest et al., 2016; Tian et al., 2016; Khanmohammadi and Chou, 2016; García-Pedrajas et al., 2012; Tian et al., 2012; García-Pedrajas and García-Osorio, 2011; Li et al., 2008). In recent years, associative memories have been extensively investigated. In (Yang et al., 2017), authors presented a nanoscale electronic device that exhibits synaptic characteristics in artificial neural networks that contributes to construct a Hopfield associative memory. In (Zhou et al., 2016), authors introduced a unified associative memory model with a novel method for designing associative memories. In (Zhong et al., 2016), authors discussed the design of a fuzzy associative memory using fuzzy correlation matrices and fuzzy clustering rules. In (Zhou et al., 2015), authors presented a novel method for designing associative memories based on discrete recurrent neural networks to accurately memorize the networks' external inputs. In (Aghajari et al., 2015), authors proposed a hetero-associative memory with dynamic behavior that can store as twice a as a regular hetero-associative memory using an extension of sparse learning method. In (Xiu et al., 2015), authors proposed an improved Hebb learning rule for an associative network to perform the associative memory of strong correlation and multi-valued sample patterns. In (Bui et al., 2015), authors proposed a novel fuzzy associative memory that stores both content and association of patterns in order to handle both erosive and dilative noises. It is worth noting that associative memories have a number of properties, including a rapid, compute efficient best-match and intrinsic noise tolerance that make them ideal for many applications. As a consequence, associative memories have emerged as a computational paradigm to efficiently solve pattern recognition tasks. Some successful applications of associative memories are the following: In (Aldape-Pérez et al., 2012b), authors presented a computational tool for engineering education, particularly it was used for hepatitis diagnosis using associative memories. In (Valle and Sussner, 2013), authors introduced quantale-based memories that generalize several lat-

**Corresponding author: Tel.: +52-55-57296000, Ext:52513;
e-mail: maldape@ipn.mx (Mario Aldape-Pérez)

tice computing approaches which can be effectively used for the storage and the recall of color images. In (López-Yáñez et al., 2014), authors proposed an associative model for time series data mining, particularly oil extraction time series were used in order to forecast monthly production as well as economic profits. In (Yáñez-Márquez et al., 2014; López-Yáñez et al., 2015), authors analyzed emerging computational tools and their impact on engineering education and computer science learning. In (Aldape-Pérez et al., 2015), authors presented a collaborative learning based associative model that was applied to pattern classification in medical datasets. In (Ferreira-Santiago et al., 2016), authors analyzed the application of associative models and the way or ways in which link prediction in social networks is able to improve teaching-learning processes in engineering education. In (Cerón-Figueroa et al., 2017), authors presented a model of pattern classification and its application to align instances from different ontologies related to e-learning.

In this paper we propose an algorithm called Smallest Normalized Difference Associative Memory (SNDAM). The paper is organized as follows. A succinct description of associative memories fundamentals is presented in Section 2. Section 3 provides a concise description of the most important characteristics of Alpha-Beta Associative Memories. In Section 5 our proposal foundations are presented. In Section 6 a brief description of non-parametric statistical analysis is presented. In Section 7 classification accuracy results achieved by each one of the compared algorithms in eight different pattern classification problems are presented. Finally, our proposal advantages, as well as some conclusions will be discussed in section 9.

2. Associative Memories

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

Table 1: Alpha Operator.

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

x	y	α(x,y)
0	0	1
0	1	0
1	0	2
1	1	1

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

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

3. Alpha-Beta Associative Memories

Alpha-Beta Associative Memories were first introduced in (Yáñez-Márquez, 2002). Alpha-Beta Associative Memories mathematical foundations are based on two binary operators: α and β. Alpha operator is used during the learning phase, while Beta operator is used during the recalling phase. The mathematical properties within these operators, allow the $\alpha\beta$ associative memories to exhibit similar characteristics to the binary version of the morphological associative memories (Ritter et al., 1998), in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall (Ritter et al., 1999, 2003). First, we define set $A = \{0, 1\}$ and set $B = \{0, 1, 2\}$, so α operator is defined as in Table 1 and β operator is defined as in Table 2. These two binary operators along with maximum ($\vee$) and minimum ($\wedge$) operators establish the mathematical tools around the Alpha-Beta model (Yáñez-Márquez and Díaz-de-León., 2003). The definitions of α and β exposed in Table 1 and Table 2 , imply that: α is increasing by the left and decreasing by the right, β is increasing by the left and right, β is the left inverse of α, see Table 3. A summary of the mathematical properties of α and β operators are shown in Table 4 and Table 5.

According to the operator that is used during the learning phase, two kinds of Alpha-Beta Associative Memories are obtained. If maximum operator ($\vee$) is used, Alpha-Beta Associative Memory *MAX* type will be obtained, denoted as $\mathbf{M}$; analogously, if minimum operator ($\wedge$) is used, Alpha-Beta Associative Memory *min* type will be obtained, denoted as $\mathbf{W}$ (Yáñez-Márquez, 2002).

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

Table 2: Beta Operator.

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

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

Table 3: Operators Properties.

$$\begin{array}{|c|}\hline \beta[\alpha(x,y),y] = x \\ \beta[\alpha(x,y),x] = x \\ \beta[\alpha(x,x),y] = y \\ \hline \end{array}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

Table 4: Alpha Operator Properties.

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

$$\begin{array}{|c|}\hline \alpha(x,x) = 1 \\ (x \leq y) \longleftrightarrow [\alpha(x,y) \leq \alpha(y,x)] \\ (x \leq y) \longleftrightarrow [\alpha(x,z) \leq \alpha(y,z)] \\ (x \leq y) \longleftrightarrow [\alpha(z,x) \geq \alpha(z,y)] \\ \hline \end{array}$$

Table 5: Beta Operator Properties.

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

$$\begin{array}{|c|}\hline \beta(1,x) = x \\ \beta(x,x) = x, \forall x \in A \\ (x \leq y) \rightarrow [\beta(x,z) \leq \beta(y,z)] \\ (x \leq y) \rightarrow [\beta(z,x) \leq \beta(z,y)] \\ \hline \end{array}$$

$$\mathbf{y} \nabla_\alpha \mathbf{x}^t = \mathbf{y} \oplus \mathbf{x}^t = \mathbf{y} \Delta_\alpha \mathbf{x}^t \tag{5}$$

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

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

3.1. Learning Phase

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

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

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

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

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

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

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

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

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

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

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

3.2. Recalling Phase

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

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

Alpha-Beta Associative Memory *MAX* type:

$$\mathbf{M}\Delta_{\beta}\mathbf{x}^{\omega} = \wedge_{j=1}^{n}\beta(v_{ij}, x_j^{\omega}) \tag{12}$$

$$\mathbf{M}\Delta_{\beta}\mathbf{x}^{\omega} = \wedge_{j=1}^{n}\beta\left(\left[\vee_{\mu=1}^{p}\alpha(y_i^{\mu}, x_j^{\mu})\right], x_j^{\omega}\right) \tag{13}$$

Alpha-Beta Associative Memory min type:

$$\mathbf{W} \triangledown_{\beta} \mathbf{x}^{\omega} = \vee_{j=1}^{n}\beta(\psi_{ij}, x_j^{\omega}) \tag{14}$$

$$\mathbf{W} \triangledown_{\beta} \mathbf{x}^{\omega} = \vee_{j=1}^{n}\beta\left(\left[\wedge_{\mu=1}^{p}\alpha(y_i^{\mu}, x_j^{\mu})\right], x_j^{\omega}\right) \tag{15}$$

Without dependence on the Alpha-Beta Associative Memory type used throughout the recalling phase, a column vector of dimension n will be obtained. A numerical example that briefly explains the operation of the algorithm is included in Section 4.

4. Numerical Example

The following shows the operation of Alpha-Beta associative memories.

Example 4.1. Let $p = 5$ be the cardinality of the fundamental set of associations, where $n = 4$ and $m = 4$ are the dimensions of input patterns and output patterns, respectively. Given the fundamental patterns $\mathbf{x}^{\mu} = \mathbf{y}^{\mu}$ $\forall\mu \in \{1, 2, ..., p\}$, obtain an Alpha-Beta Associative Memory. The fundamental associations will be denoted as: $\left\{\left(\mathbf{x}^1, \mathbf{y}^1\right), \left(\mathbf{x}^2, \mathbf{y}^2\right), ..., \left(\mathbf{x}^5, \mathbf{y}^5\right)\right\}$

$$\mathbf{x}^1 = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \quad \mathbf{x}^2 = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \quad \mathbf{x}^3 = \begin{pmatrix}0\\1\\1\\1\end{pmatrix}$$

$$\mathbf{x}^4 = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad \mathbf{x}^5 = \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

$$\mathbf{y}^1 = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \quad \mathbf{y}^2 = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \quad \mathbf{y}^3 = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

$$\mathbf{y}^4 = \begin{pmatrix}1\\1\\0\\0\end{pmatrix} \quad \mathbf{y}^5 = \begin{pmatrix}1\\0\\1\\1\end{pmatrix}$$

4.1. Learning phase

Obtain the corresponding matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_5$, according to step 1, indicated in section 3.1.

$$\mathbf{y}^1 \oplus \left(\mathbf{x}^1\right)^t = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \oplus \begin{pmatrix}1 & 1 & 0 & 1\end{pmatrix} = \begin{pmatrix}1 & 1 & 2 & 1\\1 & 1 & 2 & 1\\0 & 0 & 1 & 0\\1 & 1 & 2 & 1\end{pmatrix}$$

$$\mathbf{y}^2 \oplus \left(\mathbf{x}^2\right)^t = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \oplus \begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix} = \begin{pmatrix}2 & 2 & 2 & 1\\1 & 1 & 1 & 0\\1 & 1 & 1 & 0\\2 & 2 & 2 & 1\end{pmatrix}$$

$$\vdots$$

$$\mathbf{y}^5 \oplus \left(\mathbf{x}^5\right)^t = \begin{pmatrix}1\\0\\1\\1\end{pmatrix} \oplus \begin{pmatrix}1 & 0 & 1 & 1\end{pmatrix} = \begin{pmatrix}1 & 2 & 1 & 1\\0 & 1 & 0 & 0\\1 & 2 & 1 & 1\\1 & 2 & 1 & 1\end{pmatrix}$$

According to step 2, an Alpha-Beta Associative Memory *MAX* type denoted by $\mathbf{M}$, is obtained. Analogously, according to step 3, an Alpha-Beta Associative Memory min type denoted by $\mathbf{W}$, is obtained.

$$\mathbf{M} = \begin{pmatrix}2 & 2 & 2 & 2\\2 & 1 & 2 & 2\\2 & 2 & 1 & 1\\2 & 2 & 2 & 1\end{pmatrix} \quad ; \quad \mathbf{W} = \begin{pmatrix}1 & 1 & 1 & 1\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\1 & 0 & 1 & 1\end{pmatrix}$$

4.2. Recalling phase

Obtain the corresponding output patterns, by doing $\mathbf{M} \triangle_{\beta} \mathbf{x}^{\mu}$, $\forall\mu \in \{1, 2, ..., p\}$ as stated in expression (13) in section 3.2. The recalling results shown below are obtained using an Alpha-Beta Associative Memory *MAX* type with β min operation.

$$\mathbf{M} \triangle_{\beta} \mathbf{x}^1 = \begin{pmatrix}2 & 2 & 2 & 2\\2 & 1 & 2 & 2\\2 & 2 & 1 & 1\\2 & 2 & 2 & 1\end{pmatrix} \triangle_{\beta} \begin{pmatrix}1\\1\\0\\1\end{pmatrix} = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} = \mathbf{y}^1$$

$$\mathbf{M} \triangle_{\beta} \mathbf{x}^2 = \begin{pmatrix}2 & 2 & 2 & 2\\2 & 1 & 2 & 2\\2 & 2 & 1 & 1\\2 & 2 & 2 & 1\end{pmatrix} \triangle_{\beta} \begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} = \mathbf{y}^2$$

$$\mathbf{M} \triangle_{\beta} \mathbf{x}^3 = \begin{pmatrix}2 & 2 & 2 & 2\\2 & 1 & 2 & 2\\2 & 2 & 1 & 1\\2 & 2 & 2 & 1\end{pmatrix} \triangle_{\beta} \begin{pmatrix}0\\1\\1\\1\end{pmatrix} = \begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \mathbf{y}^3$$

$$\mathbf{M} \triangle_{\beta} \mathbf{x}^4 = \begin{pmatrix}2 & 2 & 2 & 2\\2 & 1 & 2 & 2\\2 & 2 & 1 & 1\\2 & 2 & 2 & 1\end{pmatrix} \triangle_{\beta} \begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \begin{pmatrix}1\\1\\0\\0\end{pmatrix} = \mathbf{y}^4$$

$$\mathbf{M} \triangle_{\beta} \mathbf{x}^5 = \begin{pmatrix}2 & 2 & 2 & 2\\2 & 1 & 2 & 2\\2 & 2 & 1 & 1\\2 & 2 & 2 & 1\end{pmatrix} \triangle_{\beta} \begin{pmatrix}1\\0\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\1\end{pmatrix} = \mathbf{y}^5$$

The reader can easily verify that the whole fundamental set of patterns is also completely recalled when using an Alpha-Beta Associative Memory min type with β max operation as stated in expression (15) in section 3.2.

4.3. Remarks

- Advantages

  One of the biggest advantage of Alpha-Beta Associative Memories is that this mathematical model recalls the fundamental set completely, if it is trained in auto-associative mode. This implies that all the fundamental patterns that are used along the learning phase will be retrieved without errors. The proof of the theorem that guarantees the complete recovery of the fundamental set can be found in (Yáñez-Márquez, 2002).

- Disadvantages

  The main disadvantage with this mathematical model is that it only works with binary patterns, so in case you want to work with patterns with real components, each component has to be binary coded. As a consequence, data processing complexity is increased. It should be noted that this mathematical model is very robust to additive or subtractive noise in input patterns (Yáñez-Márquez, 2002), however, this model has low classification performance when you have mixed noise in input patterns (Yáñez-Márquez, 2002). These associative memories have been widely used in many applications but the fundamental input patterns had to be coded using the Johnson-Möbius Modified Code (Yáñez-Márquez, 2002). This type of coding mantains the order relation between patterns. As a result noise type is preserved and performance is improved, but also processing complexity is increased.

5. Smallest Normalized Difference Associative Memory

In this section, Smallest Normalized Difference Associative Memory (SNDAM) theoretical foundations are presented. In order to eliminate Alpha-Beta Associative Memories disadvantages, we have to extend Alpha and Beta operators to $\mathbb{R}$ domain. Alpha operation has only one case of application; however, Beta operation has two cases according to the type of memory that is built in the training phase (*MAX* or *min*) (Alarcón-Paredes et al., 2013).

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

$$\alpha_{\mathbb{R}}(c,d) = c - d + 1 \tag{16}$$

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

$$\beta^{\vee}_{\mathbb{R}}(c,d) = \begin{cases} d - |c| - 1 & if \quad c \neq d \\ c & if \quad c = d \end{cases} \tag{17}$$

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

$$\beta^{\wedge}_{\mathbb{R}}(c,d) = \begin{cases} c - |d| - 1 & if \quad c \neq d \\ c & if \quad c = d \end{cases} \tag{18}$$

It is important to note that when an Alpha-Beta Associative Memory is trained in auto-associative mode, the main diagonal entries of the learning matrix are given by Equation (19) or explicitly, the main diagonal entries of the learning matrix has only 1's, as can be seen on matrix (20).

$$\alpha_{\mathbb{R}}(c,c) = c - c + 1 = 1, \forall c \in \mathbb{R} \tag{19}$$

$$\begin{pmatrix} 1 & \cdots & \alpha_R(x_1^{\mu}, x_n^{\mu}) \\ \vdots & \ddots & \vdots \\ \alpha_R(x_n^{\mu}, x_1^{\mu}) & \cdots & 1 \end{pmatrix} \tag{20}$$

As a consequence, complete recalling of the fundamental set is guaranteed. The proof of the theorem that guarantees that an Alpha-Beta Associative Memory recovers completely the training set, appears in (Yáñez-Márquez, 2002).

5.1. Algorithm

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

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

$$x_j^{MAX} = \bigvee_{\mu=1}^{p} x_j^{\mu} \tag{21}$$

5.1.1. Training phase

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

1. In order to build an Alpha-Beta Associative Memory *MAX* type, use Equation (22), similarly, if you want to build an Alpha-Beta Associative Memory min type, use Equation (23).

$$\mathbf{M} = \bigvee_{\mu=1}^{p} \left(\alpha_{\mathbb{R}}\left(\mathbf{x}^{\mu}, \mathbf{x}^{\mu}\right)\right) \tag{22}$$

$$\mathbf{W} = \bigwedge_{\mu=1}^{p} \left(\alpha_{\mathbb{R}}\left(\mathbf{x}^{\mu}, \mathbf{x}^{\mu}\right)\right) \tag{23}$$

   Where $\bigvee$ is the maximum operator and $\bigwedge$ is the minimum operator. After this step, we get an associative memory $\mathbf{M}$ or an associative memory $\mathbf{W}$.

2. Search for the highest absolute value of each component, using the maximum vector $\mathbf{x}^{MAX}$, as stated in Equation (21).

5.1.2. Recalling phase

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

1. Recall pattern $\mathbf{y}$ from an unknown input pattern $\tilde{\mathbf{x}}$ using Equation (24) or Equation (25):

$$\mathbf{y} = \bigwedge (\beta(\mathbf{M}, \tilde{\mathbf{x}})) \tag{24}$$

$$\mathbf{y} = \bigvee (\beta(\mathbf{M}, \tilde{\mathbf{x}})) \tag{25}$$

   Where $\bigvee$ is the maximum operator and $\bigwedge$ is the minimum operator
2. Compute the normalized difference $\boldsymbol{\delta}^{\mu}$ between the recalled pattern $\mathbf{y}$ and the fundamental input patterns $\mathbf{x}^{\mu}$, $\forall \mu \in \{1, 2, ..., p\}$ as stated in Equation (26)

$$\boldsymbol{\delta}^{\mu} = \sum_{i=1}^{n} \frac{\left| y_i - x_i^{\mu} \right|}{x_i^{max}} \tag{26}$$

3. Obtain the smallest normalized difference value $\boldsymbol{\delta}^{\mu}$ in order to identify μ
4. Use μ value to assign the class label of the pattern $\mathbf{x}^{\mu}$ to the recalled pattern $\mathbf{y}$.

6. Statistical Significance Testing

Statistical Significance Testing is useful in cases in which the main objective is to demonstrate the effectiveness of a generic learning algorithm against existing approaches. The Friedman statistic test (Friedman, 1937, 1940) is a non-parametric equivalent of the repeated-measures one-way ANOVA test. According to the null-hypothesis, which states that all the classifiers are equivalent in their performance, a rejection of this hypothesis implies the existence of differences among the performance of all the classifiers studied. The analysis is based on the ranks of each classifier on each dataset and not on the explicit performance measure (*pm*) (Demšar, 2006). The Friedman statistic test ranks the algorithms for each data set separately, the best-performing algorithm gets the rank of 1, the second best rank 2, and so on. In case of ties (like in Liver, Inflammation and Diabetes), average ranks are assigned, as shown in Table 8.

6.1. Friedman test evaluation

The Friedman test evaluation process is as follows: Consider n datasets and k classifiers to evaluate. Each algorithm is ranked for each dataset separately, according to the performance measure *pm*, from the best-performing classifier to the worst-performing classifier. Hence, for dataset S_i, the classifier f_j such that $pm_{ij} > pm_{ij'}$ $\forall j'$, j, $j' \in \{1, 2, ..., k\}$, $j \neq j$, gets the rank of 1. In the case of a d-way tie just after the rank r, assign a rank of $[(r+1) + (r+2) + \cdots + (r+d)]/d$ to each of the tied classifiers.

Definition 9. *Let R_{ij} be the rank of classifier f_j on dataset S_i. The mean rank $\overline{R}_j$ of classifier f_j on all datasets is calculated as follows:*

$$\overline{R}_j = \frac{1}{n} \sum_{i=1}^{n} R_{ij} \tag{27}$$

Definition 10. *Let R_{ij} be the rank of classifier f_j on dataset S_i. The overall mean rank $\overline{R}$ is calculated as follows:*

$$\overline{R} = \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} R_{ij} \tag{28}$$

Definition 11. *Let $\overline{R}_j$ be the mean rank of classifier f_j on all datasets and $\overline{R}$ be the overall mean rank. The Sum of Squares Total is calculated as follows:*

$$SS_{Total} = n \sum_{j=1}^{k} \left(\overline{R}_j - \overline{R} \right)^2 \tag{29}$$

Definition 12. *Let R_{ij} be the rank of classifier f_j on dataset S_i and $\overline{R}$ be the overall mean rank. The Sum of Squares Error is calculated as follows:*

$$SS_{Error} = \frac{1}{n(k-1)} \sum_{i=1}^{n} \sum_{j=1}^{k} \left(R_{ij} - \overline{R} \right)^2 \tag{30}$$

Definition 13. *Let SS_{Total} be the Sum of Squares Total (denoting the variation in the ranks) and SS_{Error} be the Sum of Squares Error (denoting the error variation). The Friedman statistic test is calculated as follows:*

$$\chi_F^2 = \frac{SS_{Total}}{SS_{Error}} \tag{31}$$

The null-hypothesis states that all the classifiers are equivalent in their performance and hence their average ranks $\overline{R}_j$ should be equal. In order to reject the null-hypothesis, the computed value χ^2, must be equal or greater than the tabled critical Chi-Square value at the prespecified level of significance (Sheskin, 2003). The χ_F^2 follows a χ^2 distribution with $k-1$ degrees of freedom (Dietterich, 1998). If the null-hypothesis is rejected, we can proceed with a post-hoc test. A post-hoc test could be used in order to find whether the proposed classifier presents statistical differences with regards to the remaining methods in the comparison (Japkowicz and Shah, 2011).

7. Experimental Phase

Throughout the experimental phase, eight datasets were used as test set to estimate the classification performance of each one of the compared algorithms. These datasets were taken from the UCI machine learning repository (Lichman, 2013), from which full documentation for all datasets can be obtained. The main characteristics of these datasets are shown in Table 6.

SNDAM performance was compared against the performance achieved by the twenty best-performing algorithms of the seventy-six available in WEKA 3: Data Mining Software in Java (Hall et al., 2009), as well as against classification accuracy results achieved by an associative model (AMBC). AMBC algorithm details appear in (Aldape-Pérez et al., 2012a). The twenty best-performing algorithms of WEKA were tested using default parameters. No parameter was modified, so the experiments can be easily replicated using WEKA, freely available on the Web (Hall et al., 2010).

Table 6: Characteristics of datasets used in the experimental phase.

	Dataset	Instances	Attributes
1.	Haberman	306	3
2.	Liver	345	6
3.	Inflammation	120	6
4.	Diabetes	768	8
5.	Breast	699	9
6.	Heart	270	13
7.	Hepatitis	155	19
8.	Parkinson	197	22

Table 9: The ten best-performing algorithms.

	Algorithm	$\overline{R}_j$
1.	Logistic	7.88
2.	MultiClassC	7.88
3.	SMO	8.38
4.	AMBC	8.75
5.	RotationForest	8.81
6.	Bagging	9.25
7.	SNDAM	10.13
8.	LMT	10.38
9.	FT	10.44
10.	SimpleLogistic	10.56

Further information on each of the algorithms that were used during the experimental phase can be found in (Aldape-Pérez et al., 2012a).

We applied the same conditions and validation schemes for each algorithm. Classification accuracy of each one of the compared algorithms was calculated using stratified 10-fold cross-validation, as suggested in (Kohavi and John, 1997).

Statistical significance testing was performed in order to demonstrate the effectiveness of Smallest Normalized Difference Associative Memory (SNDAM) against existing approaches, as suggested in (Demšar, 2006).

8. Results

In this section we analyze the classification accuracy results achieved by each one of the compared algorithms in eight different pattern classification problems.

According to the type of learning scheme, each of these can be grouped in one of the following types of classifiers: Bayesian classifiers, Functions based classifiers, Meta classifiers, Rules based classifiers, Decision Trees classifiers and Associative Models based classifiers.

The twenty-one best-performing algorithms are as follows:

- Four functions based classifiers (Logistic (le Cessie and van Houwelingen, 1992), RBFNetwork (Buhmann, 2003), SimpleLogistic (Sumner and E. Frank, 2005) and SMO (Platt, 1998)).

- Four algorithms based on the Bayesian approach (BayesNet (Christofides, 1975), NaiveBayes (John and Langley, 1995), NaiveBayesSimple (Duda and Hart, 1973) and NaiveBayesUpdateable (John and Langley, 1995)).

- Seven meta classifiers (AdaBoostM1 (Freund and Schapire, 1996), Bagging (Breiman, 1996), Dagging (Ting and Witten, 1997), MultiClassClassifier (Hall et al., 2010; Witten and Frank, 2005), RandomCommittee (Hall et al., 2010; Witten and Frank, 2005), RandomSubSpace (Ho, 1998), RotationForest (Rodriguez et al., 2006)).

- Three decision trees classifiers (FT (Landwehr et al., 2005), LMT (Landwehr et al., 2005), RandomForest (Breiman, 2001)).

- Two rules based classifiers (DecisionTable (Kohavi, 1995a) and DTNB (Hall and Frank, 2008)).

- One associative model based classifier (AMBC (Aldape-Pérez et al., 2012a)).

Table 10: Statistics of the Friedman test.

N	8
Chi-Square	24.685
df	21
Asymp. Sig.	0.261

Table 11: Critical Chi-Square values with 21 degrees of freedom (Sheskin, 2003).

p	**0.05**	0.15	0.25	0.26
χ^2	**32.671**	27.662	24.935	24.685

Table 7 shows classification accuracy achieved by each of the compared algorithms in eight different pattern classification problems in the medical field, using stratified 10-fold cross-validation, as suggested by Kohavi and John in (Kohavi, 1995b; Kohavi and John, 1997). For each compared algorithm, the values of classification accuracy averaged over all datasets are given at the end of each row. For each dataset, the highest classification accuracy is highlighted with boldface.

Classification results are as follows: Two of the four functions based classifiers (RBFNetwork (Buhmann, 2003) and SimpleLogistic (Sumner and E. Frank, 2005)) achieved the best performance in two of the eight datasets. One of the four algorithms based on the Bayesian approach (BayesNet (Christofides, 1975)) achieved the best performance in one of the eight datasets. Two of the seven meta classifiers (Bagging (Breiman, 1996), RotationForest (Rodriguez et al., 2006)) achieved the best performance in two of the eight pattern classification problems. One of the three decision trees classifiers (LMT (Landwehr et al., 2005)) achieved the best performance in two of the eight pattern classification problems. Finally, associative model based classifier (AMBC (Aldape-Pérez et al., 2012a)) achieved the best performance in three of the eight pattern classification problems.

As it is shown in Table 7 and Table 8 there is no particular method that surpasses all other algorithms in all sorts of problems. Our proposal achieved the best performance in

Table 7: Classification accuracy on 8 domains, using 10 fold cross-validation.

		Datasets								
	Algorithm	Haberman	Liver	Inflammation	Diabetes	Breast	Heart	Hepatitis	Parkinson	Average
1.	AdaBoostM1	73.20	66.66	**100.00**	74.34	95.60	82.22	67.09	85.12	80.53
2.	Bagging	73.20	**73.04**	**100.00**	74.60	96.19	83.70	69.67	87.69	82.26
3.	BayesNet	72.54	56.81	**100.00**	74.34	97.21	82.22	69.03	80.00	79.02
4.	Dagging	73.52	57.97	**100.00**	74.08	96.77	82.22	66.45	85.12	79.52
5.	DecisionTable	72.54	57.97	**100.00**	71.22	95.75	83.33	72.25	83.58	79.58
6.	DTNB	72.54	57.97	**100.00**	73.82	97.51	82.59	68.38	85.12	79.74
7.	FT	72.87	70.43	**100.00**	77.34	96.92	82.22	69.03	84.61	81.68
8.	LMT	73.85	69.85	**100.00**	**77.47**	96.48	82.22	67.09	86.15	81.64
9.	Logistic	74.50	68.69	**100.00**	77.21	96.63	83.70	68.38	86.66	81.97
10.	MultiClassC	74.50	68.69	**100.00**	77.21	96.63	83.70	68.38	86.66	81.97
11.	NaiveBayes	74.50	54.20	95.83	76.30	96.19	83.33	71.61	69.23	77.65
12.	NaiveBayesS	73.85	55.07	95.83	76.30	96.33	82.96	70.96	69.23	77.57
13.	NveBayesUpdt	74.50	54.20	95.83	76.30	96.19	83.33	71.61	69.23	77.65
14.	RndmComm	64.37	68.11	**100.00**	75.26	96.48	82.22	63.22	90.76	80.05
15.	RandomForest	67.97	70.72	**100.00**	72.39	97.07	83.70	65.16	90.76	80.97
16.	RndmSubSpace	72.22	64.05	**100.00**	75.26	95.54	82.22	67.74	88.70	80.72
17.	RBFNetwork	72.87	66.08	**100.00**	75.39	95.90	**84.07**	69.67	84.10	81.01
18.	RotationForest	73.20	**73.04**	**100.00**	76.82	97.21	82.59	66.45	90.25	82.45
19.	SimpleLogistic	73.85	71.01	**100.00**	**77.47**	96.63	82.22	66.45	84.61	81.53
20.	SMO	73.52	57.97	**100.00**	77.34	96.92	83.33	72.25	87.17	81.06
21.	AMBC	**76.33**	65.50	**100.00**	70.39	**97.80**	83.70	85.16	70.02	81.11
★	SNDAM	74.39	65.20	**100.00**	69.39	96.93	79.26	**85.81**	**93.72**	**83.09**

Table 8: Rewriting Table 7 as ranks. The ranks are used in computation of the Friedman test

		Datasets								
	Algorithm	Haberman	Liver	Inflammation	Diabetes	Breast	Heart	Hepatitis	Parkinson	$\overline{R}_j$
1.	AdaBoostM1	13.00	10.00	**10.00**	15.50	21.00	17.50	16.50	12.00	14.44
2.	Bagging	13.00	**1.50**	**10.00**	14.00	17.00	4.00	8.50	6.00	9.25
3.	BayesNet	18.00	19.00	**10.00**	15.50	3.50	17.50	10.50	18.00	14.00
4.	Dagging	10.50	16.50	**10.00**	17.00	9.00	17.50	19.00	12.00	13.94
5.	DecisionTable	18.00	16.50	**10.00**	20.00	20.00	8.50	3.50	17.00	14.19
6.	DTNB	18.00	16.50	**10.00**	18.00	2.00	12.50	13.00	12.00	12.75
7.	FT	15.00	5.00	**10.00**	3.50	7.50	17.50	10.50	14.50	10.44
8.	LMT	8.00	6.00	**10.00**	**1.50**	13.50	17.50	16.50	10.00	10.38
9.	Logistic	3.50	7.50	**10.00**	5.50	11.00	4.00	13.00	8.50	**7.88**
10.	MultiClassC	3.50	7.50	**10.00**	5.50	11.00	4.00	13.00	8.50	**7.88**
11.	NaiveBayes	3.50	21.50	21.00	9.00	17.00	8.50	5.50	21.00	13.38
12.	NaiveBayesS	8.00	20.00	21.00	9.00	15.00	11.00	7.00	21.00	14.00
13.	NveBayesUpdt	3.50	21.50	21.00	9.00	17.00	8.50	5.50	21.00	13.38
14.	RndmComm	22.00	9.00	**10.00**	12.50	13.50	17.50	22.00	2.50	13.63
15.	RandomForest	21.00	4.00	**10.00**	19.00	5.00	4.00	21.00	2.50	10.81
16.	RndmSubSpace	20.00	14.00	**10.00**	12.50	22.00	17.50	15.00	5.00	14.50
17.	RBFNetwork	15.00	11.00	**10.00**	11.00	19.00	**1.00**	8.50	16.00	11.44
18.	RotationForest	13.00	**1.50**	**10.00**	7.00	3.50	12.50	19.00	4.00	8.81
19.	SimpleLogistic	8.00	3.00	**10.00**	**1.50**	11.00	17.50	19.00	14.50	10.56
20.	SMO	10.50	16.50	**10.00**	3.50	7.50	8.50	3.50	7.00	8.38
21.	AMBC	**1.00**	12.00	**10.00**	21.00	**1.00**	4.00	2.00	19.00	8.75
★	SNDAM	6.00	13.00	**10.00**	22.00	6.00	22.00	**1.00**	**1.00**	10.13

three of the eight pattern classification problems. This should not be surprising since Wolpert and Macready (Wolpert and Macready, 1997) demonstrated that what an algorithm gains in performance on one class of problems is necessarily offset by its performance on the remaining problems.

Table 8 and Table 9 show classification accuracy as ranks. The ranks are used for Statistical Significance Testing. Table 10 shows Statistics of the Friedman test for the twenty-two algorithms in 8 different domains.

In order to reject the null hypothesis, the computed value χ^2_r must be equal or greater than the tabled critical Chi-Square value at the prespecified level of significance p. The computed Chi-Square value is evaluated with the Chi-Square Distribution (Sheskin, 2003).

For 21 degrees of freedom (df), the tabled χ^2_{95} value (which is the Chi-Square value at the 95th percentile) is employed as the 0.05 critical p value. Table 11 shows Critical Chi-Square values with 21 degrees of freedom.

For $df = 21$, the tabled critical 0.05 Chi-Square value is $\chi^2_{95} = 32.671$. Since the computed value $\chi^2_r = 24.685$ is smaller than $\chi^2_{95} = 32.671$, the null hypothesis is not rejected. As a consequence, there is no need to proceed with a post-hoc test.

9. Conclusion

In this paper, a novel approach to perform pattern classification tasks is presented. This model is called Smallest Normalized Difference Associative Memory (SNDAM).

After the analysis of results shown in Table 7, Table 8 and Table 9, we can say that the proposed algorithm has competitive performance compared against the performance achieved by the twenty best-performing algorithms of the seventy-six available in WEKA 3: Data Mining Software in Java (Hall et al., 2009), as well as against classification accuracy results achieved by an associative model (AMBC) (Aldape-Pérez et al., 2012a).

The null-hypothesis states that all the classifiers are equivalent in their performance and hence their average ranks $\overline{R}_j$ should be equal. In order to reject the null-hypothesis, the computed value χ^2_r, must be equal or greater than the tabled χ^2_{95} value at the prespecified level of significance.

Since the null-hypothesis was not rejected, we can conclude that the proposed classifier does not present statistical differences with regards to the twenty-one remaining methods in the comparison.

The results obtained through the Statistical Significance Testing indicate that the proposed algorithm is competitive. Classification performance achieved by the proposed algorithm is equivalent to that achieved by the other twenty-one methods of the current literature.

Experimental results have shown that SNDAM achieved the best performance in three of the eight pattern classification problems, using 10-fold cross-validation, as shown in Table 7.

It should be noted that our proposal achieved the best classification accuracy averaged over all datasets.

Statistical significance tests show that associative memories can classify medical patterns, efficiently.

The results presented in this paper demonstrate associative memories potential for pattern classification systems.

Acknowledgements. The authors of the present paper would like to thank the following institutions for their economical support to develop this work: Science and Technology National Council of Mexico (CONACYT), SNI, National Polytechnic Institute of Mexico (COFAA, SIP, CIDETEC, and CIC). The authors also would like to thank Dr. Yenny Villuendas-Rey and Dr. Abril Uriarte-Arcia for the help and guidance in the statistical tests and for her valuable comments.

References

Aghajari, Z., Teshnehlab, M., Motlagh, M.J., 2015. A novel chaotic hetero-associative memory. Neurocomputing 167, 352 – 358.

Alarcón-Paredes, A., Pogrebnyak, O., Argüelles-Cruz, A.J., 2013. Image transform based on alpha-beta associative memories. Computación y Sistemas 17, 527–541.

Aldape-Pérez, M., Yáñez-Márquez, C., Camacho-Nieto, O., Argüelles-Cruz, A.J., 2012a. An associative memory approach to medical decision support systems. Computer Methods and Programs in Biomedicine 106, 287–307.

Aldape-Pérez, M., Yáñez-Márquez, C., Camacho-Nieto, O., Argüelles-Cruz, A.J., 2012b. A new tool for engineering education: Hepatitis diagnosis using associative memories. International Journal of Engineering Education 28, 1399–1405.

Aldape-Pérez, M., Yáñez-Márquez, C., Camacho-Nieto, O., López-Yáñez, I., Argüelles-Cruz, A.J., 2015. Collaborative learning based on associative models: Application to pattern classification in medical datasets. Computers in Human Behavior 51, 771–779.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Buhmann, M.D., 2003. Radial Basis Functions: Theory and Implementations (Cambridge Monographs on Applied and Computational Mathematics). Cambridge University Press.

Bui, T.D., Nong, T.H., Dang, T.K., 2015. Improving learning rule for fuzzy associative memory with combination of content and association. Neurocomputing 149, Part A, 59 – 64.

Cerón-Figueroa, S., López-Yáñez, I., Alhalabi, W., Camacho-Nieto, O., Villuendas-Rey, Y., Aldape-Pérez, M., Yáñez-Márquez, C., 2017. Instance-based ontology matching for e-learning material using an associative pattern classifier. Computers in Human Behavior 69, 218 – 225.

le Cessie, S., van Houwelingen, J., 1992. Ridge estimators in logistic regression. Applied Statistics 41, 191–201.

Christofides, N., 1975. Graph Theory: An Algorithmic Approach (Computer Science and Applied Mathematics). Academic Press, Inc., Orlando, FL, USA.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1923.

Duda, R., Hart, P., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Ferreira-Santiago, Á., Yáñez-Márquez, C., López-Yáñez, I., Camacho-Nieto, O., Aldape-Pérez, M., Argüelles-Cruz, A.J., 2016. Enhancing engineering education through link prediction in social networks. International Journal of Engineering Education 32, 1566–1578.

Freund, Y., Schapire, R., 1996. Experiments with a new boosting algorithm, in: Kaufmann, M. (Ed.), Thirteenth International Conference on Machine Learning, San Francisco. pp. 148–156.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association 32, 675–701.

Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11, 86–92.

García-Pedrajas, N., García-Osorio, C., 2011. Constructing ensembles of classifiers using supervised projection methods based on misclassified instances. Expert Systems with Applications 38, 343 – 359.

García-Pedrajas, N., Maudes-Raedo, J., García-Osorio, C., Rodríguez-Díez, J.J., 2012. Supervised subspace projections for constructing ensembles of classifiers. Information Sciences 193, 1 – 21.

Hall, M., Frank, E., 2008. Combining naive bayes and decision tables, in:

Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS), AAAI press. pp. 318–319.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: An update. SIGKDD Explorations 11, 10–18.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2010. Weka 3: Data mining software in java.

Ho, T.K., 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 832–844.

Japkowicz, N., Shah, M., 2011. Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press.

John, G., Langley, P., 1995. Estimating continuous distributions in bayesian classifiers, in: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, Morgan Kaufmann. pp. 338–345.

Khanmohammadi, S., Chou, C.A., 2016. A gaussian mixture model based discretization algorithm for associative classification of medical data. Expert Systems with Applications 58, 119 – 129.

Kohavi, R., 1995a. The power of decision tables, in: 8th European Conference on Machine Learning, Springer. pp. 174–189.

Kohavi, R., 1995b. A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 95), pp. 1137–1145.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artificial Intelligence 97, 273–324.

Landwehr, N., Hall, M., Frank, E., 2005. Logistic model trees. Machine Learning 59, 161–205.

Li, M., Tian, J., Chen, F., 2008. Improving multiclass pattern recognition with a co-evolutionary {RBFNN}. Pattern Recognition Letters 29, 392 – 406.

Lichman, M., 2013. UCI machine learning repository.

López-Yáñez, I., Sheremetov, L., Yáñez-Márquez, C., 2014. A novel associative model for time series data mining. Pattern Recognition Letters 41, 23–33.

López-Yáñez, I., Yáñez-Márquez, C., Camacho-Nieto, O., Aldape-Pérez, M., Argüelles-Cruz, A.J., 2015. Collaborative learning in postgraduate level courses. Computers in Human Behavior 51, 938–944.

Platt, J., 1998. Fast training of support vector machines using sequential minimal optimization. Advances in Kernel Methods-Support Vector Learning, MIT Press.

Ritter, G.X., Díaz-de-León, J.L., Sussner, P., 1999. Morphological bidirectional associative memories. Neural Networks 12, 851–867.

Ritter, G.X., Sussner, P., Díaz-de-León., J.L., 1998. Morphological associative memories. IEEE Trans. Neural Networks 9, 281–293.

Ritter, G.X., Urcid, G., Iancu, L., 2003. Reconstruction of patterns from noisy inputs using morphological associative memories. Journal of Mathematical Imaging and Vision 19, 95–111.

Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J., 2006. Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1619–1630.

Sheskin, D.J., 2003. Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press Taylor & Francis Group. 3rd edition.

Steinbuch, K., 1961. Die lernmatrix. Kybernetik 1, 36–45.

Steinbuch, K., 1964. Adaptive networks using learning matrices. Kybernetik 2, 148–152.

Steinbuch, K., Frank, H., 1961. Nichtdigitale lernmatrizen als perzeptoren. Kybernetik 1, 117–124.

Sumner, M., E. Frank, M.H., 2005. Speeding up logistic model tree induction, in: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Springer. pp. 675–683.

Tian, J., Li, M., Chen, F., Feng, N., 2016. Learning subspace-based rbfnn using coevolutionary algorithm for complex classification tasks. IEEE Trans. Neural Networks and Learning Systems 27, 47 – 61.

Tian, J., Li, M., Chen, F., Kou, J., 2012. Coevolutionary learning of neural network ensemble for complex classification tasks. Pattern Recognition 45, 1373 – 1385.

Ting, K.M., Witten, I.H., 1997. Stacking bagged and dagged models, in: Fisher, D.H. (Ed.), Fourteenth international Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA. pp. 367–375.

Valle, M.E., Sussner, P., 2013. Quantale-based autoassociative memories with an application to the storage of color images. Pattern Recognition Letters 34, 1589 – 1601.

Verbiest, N., Vluymans, S., Cornelis, C., Garca-Pedrajas, N., Saeys, Y., 2016. Improving nearest neighbor classification using ensembles of evolutionary generated prototype subsets. Applied Soft Computing 44, 75 – 88.

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1, 67–82.

Xiu, C., Liu, C., Cheng, Y., 2015. Associative memory network and its hardware design. Neurocomputing 158, 204 – 209.

Yáñez-Márquez, C., 2002. Associative Memories based on Order Relations and Binary Operators (In Spanish). Ph.D. thesis. Instituto Politécnico Nacional.

Yáñez-Márquez, C., Aldape-Pérez, M., López-Yáñez, I., Camacho-Nieto, O., 2014. Emerging computational tools: Impact on engineering education and computer science learning. International Journal of Engineering Education 30, 533–542.

Yáñez-Márquez, C., Díaz-de-León., J.L., 2003. Associative memories based on order relations and binary operators. Computación y Sistemas 6.

Yang, J., Wang, L., Wang, Y., Guo, T., 2017. A novel memristive hopfield neural network with application in associative memory. Neurocomputing 227, 142 – 148.

Zhong, C., Pedrycz, W., Li, Z., Wang, D., Li, L., 2016. Fuzzy associative memories: A design through fuzzy clustering. Neurocomputing 173, Part 3, 1154 – 1162.

Zhou, C., Zeng, X., Jiang, H., Han, L., 2015. A generalized bipolar auto-associative memory model based on discrete recurrent neural networks. Neurocomputing 162, 201 – 208.

Zhou, C., Zeng, X., Yu, J., Jiang, H., 2016. A unified associative memory model based on external inputs of continuous recurrent neural networks. Neurocomputing 186, 44 – 53.