Delta Associative Memory: An Efficient Pattern Classifier

Mario Aldape-Pérez, Cornelio Yáñez-Márquez
and Amadeo José Argüelles-Cruz

Center for Computing Research, CIC
National Polytechnic Institute, IPN
Mexico City, Mexico
maldape@ieee.org; cyanez@cic.ipn.mx; jamadeo@cic.ipn.mx
http://www.aldape.org.mx

Abstract. The *Linear Associator* is one of the classical models of associative memories that can easily work as a binary pattern classifier if output patterns are appropriately chosen. However, this mathematical model undergoes fundamental patterns misclassification whenever fundamental input pattern cross-talk influence occurs. In this paper, a novel algorithm that overcomes *Linear Associator* weaknesses is proposed. The proposed algorithm computes a differential associative memory in order to obtain a dynamic threshold value for each unknown pattern to be classified. The efficiency and effectiveness of our approach is demonstrated through comparisons with other methods using real-world data.

Keywords: Associative Memories, Linear Associator, Machine Learning, Pattern Recognition.

1 Introduction

Pattern Recognition has existed for many years in a wide range of human activity, however, the general pattern recognition problem can be stated in the following form: Given a collection of objects belonging to a predefined set of classes and a set of measurements on these objects, identify the class of membership of each one of these objects by a suitable analysis of their measurements (features) [1]. At present, Pattern Recognition comprises a vast body of methods supporting the development of numerous applications in many different domains [2]. From an information theoretical perspective, an associative memory can be regarded as an extension of the neural computing approach for pattern recognition [3]. Furthermore, 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. The majority of these applications are focused on problems related to pattern recognition tasks where the system takes in a pattern and emits another pattern [4]. As a consequence, associative memories have emerged as a practical technology with successful applications in countless pattern recognition tasks [5].

*Received 17/03/09
Accepted 27/04/09
Final version 07/05/09*

The *Linear Associator*, which is one of the classical models of associative memories, is a heteroassociative memory that can easily work as a binary pattern classifier if output patterns are appropriately chosen. However, this mathematical model undergoes fundamental patterns misclassification whenever fundamental input pattern cross-talk influence occurs. In this paper, a novel algorithm that overcomes *Linear Associator* weaknesses is proposed. The experimental outcomes showed that fundamental input pattern cross-talk influence can be annulled by means of a dynamic threshold value which is computed for each unknown input pattern to be classified. As a consequence, classification rate is improved.

In the following section, a brief description of associative memories fundamentals is presented. In Section 3, *Linear Associator* technical details are presented. In Section 4, Delta Associative Memory mathematical foundations are presented. In Section 5, the main characteristics of the datasets that were used along the experimental phase are presented. A brief description of the algorithms used for comparison along the experimental phase is presented in Section 6 while in Section 7 some experimental results are shown using real-world data. Delta Associative Memory advantages, as well as a short conclusion will be discussed in Section 8.

2 Associative Memories

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

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

A distorted version of a pattern $\mathbf{x}^k$ to be recalled will be denoted as $\widetilde{\mathbf{x}}^k$. An unknown input pattern to be recalled will be denoted as $\mathbf{x}^\omega$. If an unknown input pattern $\mathbf{x}^\omega$ with $\omega \in \{1, 2, ..., k, ..., p\}$ is fed to an associative memory $\mathbf{M}$, it happens that the output corresponds exactly to the associated pattern $\mathbf{y}^\omega$, it is said that recalling is correct [6].

3 Linear Associator

It is worth pointing out that James A. Anderson and Teuvo Kohonen obtained amazingly similar results known nowadays as *Linear Associator* [4,7].

3.1 Learning Phase

Let $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$.be the fundamental set. In order to obtain an associative memory $\mathbf{M}$, the *Learning Phase* is done in two stages:

1. Consider each one of the p associations $(\mathbf{x}^\mu, \mathbf{y}^\mu)$, so an $m \times n$ matrix is obtained by $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$

$$\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, ..., x_n^\mu) = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \cdots & y_1^\mu x_j^\mu & \cdots & y_1^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \cdots & y_i^\mu x_j^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \cdots & y_m^\mu x_j^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix} \quad (1)$$

2. An associative memory $\mathbf{M}$ is obtained by adding all the p matrices

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \quad (2)$$

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

$$m_{ij} = \sum_{\mu=1}^{p} y_i^\mu x_j^\mu \quad (3)$$

3.2 Recalling Phase

The *Recalling Phase* for the *Linear Associator* is done by operating an associative memory $\mathbf{M}$ with an unknown input pattern $\mathbf{x}^\omega$, where $\omega \in \{1, 2, ..., k, ..., p\}$. Operate $\mathbf{M} \cdot \mathbf{x}^\omega$ as follows:

$$\mathbf{M} \cdot \mathbf{x}^\omega = \left[\sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t \right] \cdot \mathbf{x}^\omega \quad (4)$$

Lets expand expression (4), which is:

$$\mathbf{M}\cdot\mathbf{x}^{\omega}=\left[\mathbf{y}^{1}\cdot\left(\mathbf{x}^{1}\right)^{t}+\mathbf{y}^{2}\cdot\left(\mathbf{x}^{2}\right)^{t}+\cdots+\mathbf{y}^{\omega}\cdot\left(\mathbf{x}^{\omega}\right)^{t}+\cdots+\mathbf{y}^{p}\cdot\left(\mathbf{x}^{p}\right)^{t}\right]\cdot\mathbf{x}^{\omega}$$
$$\mathbf{M}\cdot\mathbf{x}^{\omega}=\left[\mathbf{y}^{1}\cdot\left(\mathbf{x}^{1}\right)^{t}\right]\cdot\mathbf{x}^{\omega}+\cdots+\left[\mathbf{y}^{\omega}\cdot\left(\mathbf{x}^{\omega}\right)^{t}\right]\cdot\mathbf{x}^{\omega}+\cdots+\left[\mathbf{y}^{p}\cdot\left(\mathbf{x}^{p}\right)^{t}\right]\cdot\mathbf{x}^{\omega}$$
$$\mathbf{M}\cdot\mathbf{x}^{\omega}=\mathbf{y}^{1}\cdot\left[\left(\mathbf{x}^{1}\right)^{t}\cdot\mathbf{x}^{\omega}\right]+\cdots+\mathbf{y}^{\omega}\cdot\left[\left(\mathbf{x}^{\omega}\right)^{t}\cdot\mathbf{x}^{\omega}\right]+\cdots+\mathbf{y}^{p}\cdot\left[\left(\mathbf{x}^{p}\right)^{t}\cdot\mathbf{x}^{\omega}\right]$$

We obtain:

$$\mathbf{M}\cdot\mathbf{x}^{\omega}=\mathbf{y}^{\omega}\cdot\left[\left(\mathbf{x}^{\omega}\right)^{t}\cdot\mathbf{x}^{\omega}\right]+\sum_{\mu\neq\omega}\mathbf{y}^{\mu}\cdot\left[\left(\mathbf{x}^{\mu}\right)^{t}\cdot\mathbf{x}^{\omega}\right]\tag{5}$$

Expression (5) let us know about which restrictions have to be observed thus correct recalling is achieved. These restrictions are:

a) $\left[\left(\mathbf{x}^{\omega}\right)^{t}\cdot\mathbf{x}^{\omega}\right]=1$

b) $\left[\left(\mathbf{x}^{\mu}\right)^{t}\cdot\mathbf{x}^{\omega}\right]=0$ whenever $\mu\neq\omega$

Given an arbitrary chosen index ω , $\forall\omega\in\{1,2,...,k,...,p\}$, means that input pattern $\mathbf{x}^{\mu}$ should be orthonormal. This restriction is expressed as:

$$\left(\mathbf{x}^{\mu}\right)^{t}\cdot\mathbf{x}^{\omega}=\begin{cases}1 \text{ if } \mu=\omega\\ \\ 0 \text{ if } \mu\neq\omega\end{cases}\tag{6}$$

If condition (6) is met, then a correct recalling is expected [6]. Therefore, expression (5) is expressed as:

$$\mathbf{M}\cdot\mathbf{x}^{\omega}=\mathbf{y}^{\omega}.$$

Nevertheless if orthonormality condition is not met, two situations appear:

· Factor $\left[\left(\mathbf{x}^{\omega}\right)^{t}\cdot\mathbf{x}^{\omega}\right]$ is not equal to 1

· Term $\sum_{\mu\neq\omega}\mathbf{y}^{\mu}\cdot\left[\left(\mathbf{x}^{\mu}\right)^{t}\cdot\mathbf{x}^{\omega}\right]$ is not equal to 0

This term is known as *cross-talk*, it represents some kind of noise that comes from input patterns interaction [4,7]. As a consequence correct recalling is not achieved, except in those cases where the number of stored patterns is rather small compared to n [5,6].

4 Delta Associative Memory

In this section, a novel algorithm that overcomes *Linear Associator* weaknesses is proposed. Due to the fact that an order relation between patterns implies an order relation between their characteristic set and vice versa [11], *cross-talk* influence can be annulled by means of a dynamic threshold value which is computed for each unknown input pattern to be classified. *Delta Associative Memory* algorithm applies the same learning phase as the *Linear Associator*, while a completely different recalling phase is proposed.

In what follows, let $\mathbf{M}$ be an associative memory whose ij-th component is denoted by m_{ij} and let ω be an index such that $\omega \in \{1,2,...,k,...,p\}$. Let $\mathbf{x}^{\omega} \in \mathbb{R}^n$ be an unknown input pattern to be classified and let $m, n \in \mathbb{Z}^+$ be the dimension of the output patterns and input patterns, respectively.

Definition 1. ***Differential Associative Memory.*** *A Differential Associative Memory is denoted by Ψ^{ω}. Therefore, the ij-th component of Ψ^{ω}, denoted by ψ_{ij}^{ω}, is obtained according to the following rule:*

$$\psi_{ij}^{\omega} = |m_{ij} - x_j^{\omega}| \tag{7}$$

$\forall i \in \{1,2,...,m\}$, $\forall j \in \{1,2,...,n\}$.

Definition 2. ***Maximum threshold value.*** *The maximum threshold value, denoted by ζ^{ω}, is obtained according to the following rule:*

$$\zeta^{\omega} = \bigvee_{i=1}^{m} \left[\bigvee_{j=1}^{n} [\psi_{ij}^{\omega}] \right] \tag{8}$$

where $\bigvee$ is the maximum operator.

Definition 3. ***Minimum threshold value.*** *The minimum threshold value, denoted by α^{ω}, is obtained according to the following rule:*

$$\alpha^{\omega} = \bigwedge_{i=1}^{m} \left[\bigwedge_{j=1}^{n} [\psi_{ij}^{\omega}] \right] \tag{9}$$

where $\bigwedge$ is the minimum operator.

Definition 4. ***Delta Associative Memory (DAM).*** *Let θ^{ω} be the dynamic threshold value, such that $\alpha^{\omega} \leq \theta^{\omega} \leq \zeta^{\omega}$. A Delta Associative Memory is denoted by Δ^{ω}. Therefore, the ij-th component of Δ^{ω}, denoted by δ_{ij}^{ω}, is obtained according to the following rule:*

$$\delta_{ij}^{\omega} = \begin{cases} 1 & \text{if } \ |m_{ij} - x_j^{\omega}| \leq \theta^{\omega} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Definition 5. ***Positive contributions vector.*** *The Positive contributions vector is denoted by σ^{ω}. Therefore, the i-th component of σ^{ω}, denoted by σ_i^{ω}, is obtained according to the following rule:*

$$\sigma_i^{\omega} = \sum_{j=1}^{n} \delta_{ij}^{\omega} \tag{11}$$

Definition 6. ***Transition vector.*** *The Transition vector is denoted by τ^{ω}. Therefore, the i-th component of τ^{ω}, denoted by τ_i^{ω}, is obtained according to the following rule:*

$$\tau_i^{\omega} = \begin{cases} 1 & \text{if } \ \sigma_i^{\omega} = \bigvee_{h=1}^{m} \left[\sigma_h^{\omega}\right] \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $\bigvee$ is the maximum operator.

Definition 7. ***Unambiguously recalled class vector.*** *The Unambiguously recalled class vector, denoted by y^{ω}, is obtained according to the following rule:*

$$y^{\omega} = \begin{cases} \tau^{\omega} & \text{if } \ \sum_{i=1}^{m} \tau_i^{\omega} \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

4.1 Learning Phase

Generate a matrix $\mathbf{M}$ that will store the p associations of the fundamental set $\{(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^p, \mathbf{y}^p)\}$, where $\mathbf{x}^{\mu} \in \mathbb{R}^n$ and $\mathbf{y}^{\mu} \in A^m$ $\forall \mu \in \{1, 2, ..., p\}$. It is worth pointing out that there are m different classes. Therefore, each one of the input patterns belongs to a k class, $k \in \{1, 2, ..., m\}$, represented by a column vector $\mathbf{y}^{\mu}$, whose components will be assigned by $y_k^{\mu} = 1$, so $y_j^{\mu} = 0$ for $j = 1, 2..., k-1, k+1, ...m$; hence, the class statements are given in a 1-out-of-m-code [11], also known as *one-hot* codification [5,6].

```
Given:
    The fundamental set of associations $\{(\mathbf{x}^{\mu}, \mathbf{y}^{\mu}) \mid \mu = 1, 2, ..., p\}$ with
$p$ as the cardinality of the set
Algorithm:
    Obtain $p$ matrices according to expression (1).
    for $\mu = 1$ to $p$ do
    {
        for $i = 1$ to $m$ do
        {
            for $j = 1$ to $n$ do
            {
            Compute $m_{ij}$ using expression (2).
            }
        }
    }
```

4.2 Classification Phase

Consists of finding the class which an unknown input pattern $\mathbf{x}^{\omega} \in \mathbb{R}^n$ belongs to. Finding the class means getting $\mathbf{y}^{\omega} \in A^m$ that corresponds to $\mathbf{x}^{\omega}$. If when an unknown input pattern $\mathbf{x}^{\omega}$ is fed to an associative memory $\mathbf{M}$, it happens that the output corresponds exactly to the associated pattern $\mathbf{y}^{\omega}$, it is said that classification is correct.

```
Given:
    An unknown input pattern $x^\omega \in \mathbb{R}^n$
Algorithm:
    Obtain a Differential Associative Memory $\Psi^\omega$, using (7)
    Compute the maximum threshold value $\zeta^\omega$, using (8)
    Compute the minimum threshold value $\alpha^\omega$, using (9)
    Initialize the dynamic threshold value $\theta^\omega$, that is, $\theta^\omega = \alpha^\omega$
    While $[(\sum_{i=1}^{m} \tau_i^\omega > 1)$ and $(\theta^\omega < \zeta^\omega)]$ do
    {
        Obtain a Delta Associative Memory $\Delta^\omega$, using (10)
        Compute the positive contributions $\sigma^\omega$, using (11)
        Compute the transition vector $\tau^\omega$, using (12)
        Assign the recalled class vector $y^\omega$, using (13)
        Increment the dynamic threshold value $\theta^\omega$, that is, $\theta^\omega = \theta^\omega + 1$
    }
    Assign the unambiguously recalled class vector $y^\omega$, using (13)
```

5 Machine Learning Databases

Nowadays, the University of California Machine Learning Repository maintains 177 different datasets which can be used as test benches in order to obtain a fair performance comparison between different algorithms. In this section, a brief description of the datasets that were used along the experimental phase is presented.

5.1 Iris Data Set

This is perhaps the best known database to be found in the pattern recognition literature [8-10]. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant (Iris Setosa, Iris Versicolour, and Iris Virginica). One class is linearly separable from the other two; the latter are not linearly separable from each other. Each instance is conformed by 4 numerical features and a class tag.

5.2 Pima Indians Diabetes Database

This database was originally owned by the National Institute of Diabetes and Digestive and Kidney Diseases. This dataset was used to forecast the onset of diabetes mellitus. The diagnostic was investigated according to the World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care). Each instance is conformed by 8 numerical features and a class tag.

5.3 Wisconsin Breast Cancer Database

This database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. This dataset was assembled using clinical reports that arrive periodically. This database has been widely used for medical diagnosis applied to breast cytology. Each instance has one of 2 possible classes: benign or malignant. Each instance is conformed by 9 numerical features and a class tag.

5.4 Heart Disease Database

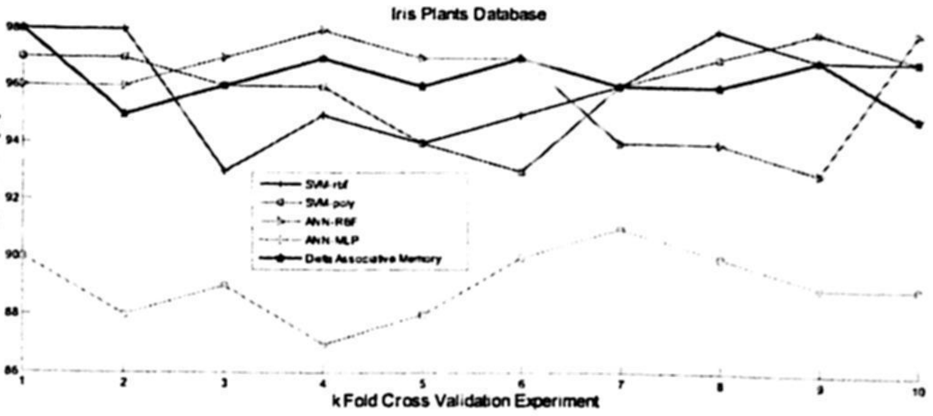
This database has been widely used in heart disease diagnosis. The data was collected by Robert Detrano, M.D. from V.A. Medical Center, Long Beach and Cleveland Clinic Foundation. The original dataset contains 76 attributes, but all published experiments refer to using a subset of 14 of them (13 numerical features and a class tag). The main purpose of this dataset is to forecast the presence of heart disease in the patient.

6 Algorithms Compared

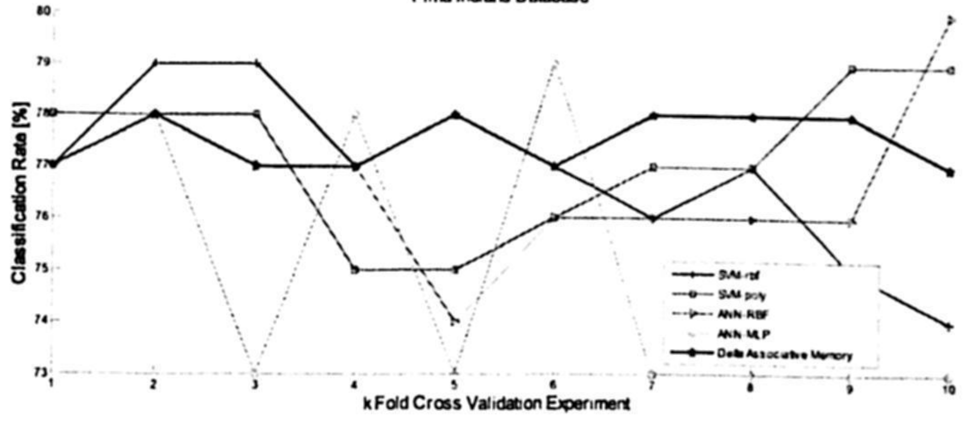
Four of the most commonly used classification models were tested along the experimental phase. Support Vector Machines using polynomial kernel (SVMpoly) and radial basis kernel (SVMrbf) were tested using the OSU-SVM Matlab Toolbox Version 3.00 obtained from [15]. For SVMrbf, γ was tuned from the possible values $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ while the constrain penalty C was tuned from the values $\{10^{-1}, 1, 10, 10^{2}, 10^{3}\}$. The SVMpoly uses a degree from the range 2 to 6 and constrain penalty C from the set $\{0.05, 0.1, 0.5, 1, 5\}$. Two types of neural networks, the radial-basis network (ANNrbf) and the multi-layered perceptron (ANNmlp) were tested using the Matlab Toolbox implementation. The limit to the number of hidden neurons added by Matlab in training the ANNrbf was set to the lower of the number of instances and ten times the number of classes. The only parameter tuned was the spread of the transfer function, from 0.1 to 1.0 in increments of 0.1.

7 Experimental Results

Throughout the experimental phase, four databases taken from the UCI Machine Learning Repository were used (http://archive.ics.uci.edu/ml/). The main characteristics of these data sets have been expounded in section 5 and are summarized in Table 1. The experiments have been carried out as follows: Firstly, in order to obtain an associative memory **M**, the same number of input vectors for each class was randomly taken, which means that a balanced classifier is guaranteed [12]. Afterwards, in order to obtain reasonably unbiased performance estimates, the dataset was broken into k partitions (in our case $k = 10$). **Delta Associative Memory** behavior was evaluated using Stratified k Fold Cross Validation technique; as a result, the classification performance estimates are reasonably unbiased [12-14]. The classification performance for each one of the compared algorithms is shown using Cartesian coordinates in two dimensions (Figure 1 to Figure 4). The horizontal axis represents the k-th partition of the fundamental set that was evaluated; while the vertical axis represents the classification rate achieved for each one of the compared algorithms over the k-th partition of the fundamental set. Efficiency of the developed method was tested by performing the experiments 20 times, each time with different number of randomly taken input vectors. Performance of the classification phase is measured in terms of error rate [11], in this way, classifier accuracy is estimated through classification rate over unseen patterns [11-14].



Averaged classification results: Iris Plants Database



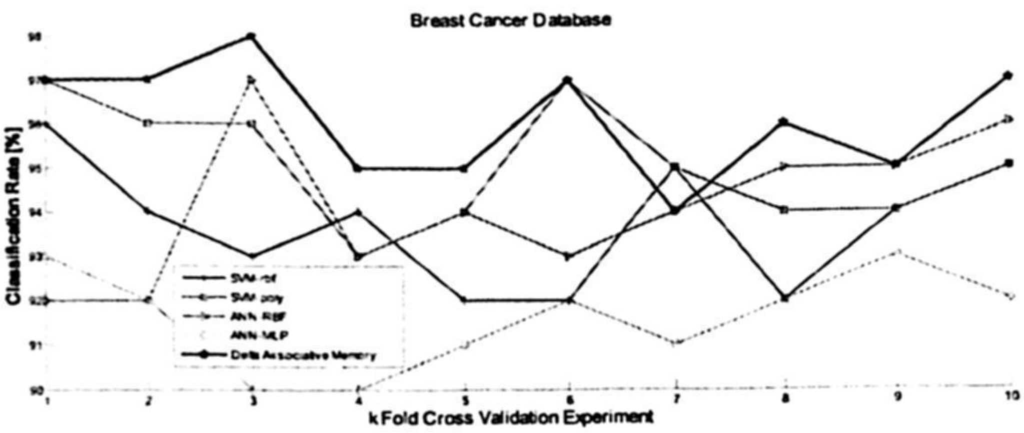
Averaged classification results: Pima Indians Database

Table 1. Some characteristics of the data sets used in the experiments

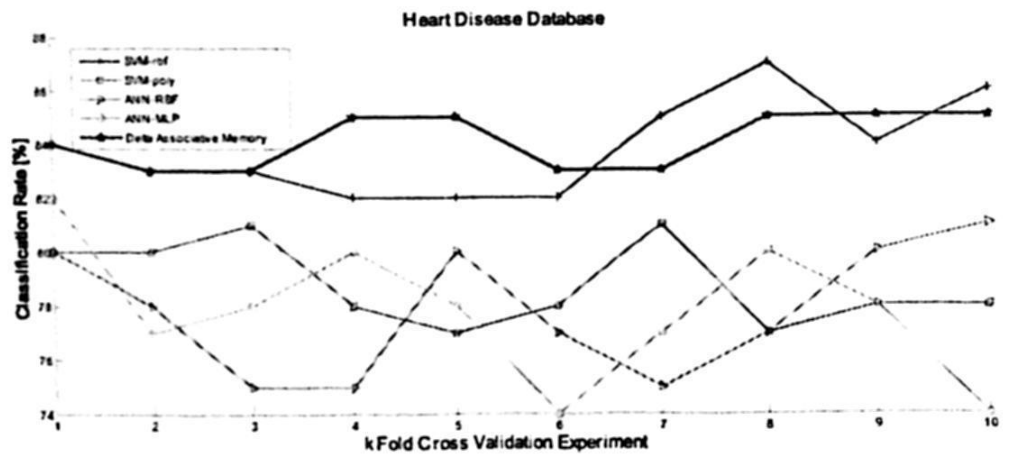
	Iris	Pima	Breast	Heart
Features	4	8	9	13
Classes	3	2	2	2
Patterns	150	768	699	270

Table 2. Mean Classification Rates

	SVM-rbf	SVM-poly	ANN-RBF	ANN-MLP	Delta AM
Iris	96.19%	96.19%	96.03%	89.18%	**96.27%**
Pima	76.97%	77.32%	76.91%	75.03%	**77.43%**
Breast	93.89%	95.16%	94.16%	91.64%	**96.08%**
Heart	83.85%	78.93%	77.78%	77.81%	**84.19%**



Averaged classification results: Breast Cancer Database



Averaged classification results: Heart Disease Database

8 Conclusions and Ongoing Research

In this paper a new algorithm which is based on one of the classical models of associative memories has been introduced. *Linear Associator* weaknesses are solved by means of a dynamic threshold value which is computed for each unknown input pattern to be classified. The algorithm computes a differential associative memory in order to obtain the maximum and minimum threshold values which represent the dynamic threshold upper and lower bounds, respectively. As a direct consequence, classification error rate is improved. Experimental results have shown that this algorithm is an efficient way to improve classifier accuracy.

Currently, we are exploring how to use the proposed approach for feature selection in machine learning and data mining. We are also working on a Xilinx Virtex II Pro FPGA implementation, based on recent mathematical results.

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

References

1. Yáñez-Márquez, C., Felipe-Riverón, E. M., López-Yáñez, I., & Flores-Carapia, R. (2006). A Novel Approach to Automatic Color Matching. Lecture Notes in Computer Science (LNCS), 4225, 529-538.
2. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I. (2006). Alpha-Beta Bidirectional Associative Memories Based Translator. International Journal of Computer Science and Network Security (IJCSNS), 6, 190-194.
3. Yáñez-Márquez, C., Sánchez-Fernández, L. P., & López-Yáñez, I. (2006). Alpha-Beta Associative Memories for Gray Level Patterns. Lecture Notes in Computer Science (LNCS), 3971, 818-823.
4. Kohonen, T. (1972). Correlation Matrix Memories. IEEE Transactions on Computers, 21, 353-359.
5. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I. (2006). A New Model of BAM: Alpha-Beta Bidirectional Associative Memories. Lecture Notes in Computer Science (LNCS), 4263, 286-295.
6. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., & López-Yáñez, I. (2007). Alpha–Beta bidirectional associative memories: theory and applications. Neural Processing Letters, 26, 1-40.
7. Anderson, J.A., & Rosenfeld, E. (1990). Neurocomputing: Fundations of Research, Cambridge: MIT Press.
8. Márques de Sá, J.P. (2001). Pattern Recognition, Concepts, Methods and Application. Springer, Germany.
9. Webb, A. (1999). Statistical Pattern Recognition. Oxford University Press, USA.
10. Jain, A.K., Duin, R.P.W., & Jianchang Mao. (2000). Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, 1, 4-37.
11. Aldape-Pérez, M., Román-Godínez, I., & Camacho-Nieto, O. (2008). Thresholded Learning Matrix for Efficient Pattern Recalling. Lecture Notes In Computer Science (LNCS), 5197, 445-452.

12. Pal, S., & Mitra, S. (1999). Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing . John Wiley & Sons, USA.
13. Hassoun, M.H. (1995). Fundamentals of Artificial Neural Networks. MIT Press, Cambridge.
14. Ritter, G. X., & Sussner, P. (1996). An Introduction to Morphological Neural Networks. in Proceedings of the 13th International Conference on Pattern Recognition, vol. IV, Track D 709-717.
15. OSU SVM Classifier Matlab Toolbox (ver 3.00) at http://www.ece.osu.edu/~maj/osu_svm/