Comparative Analysis of Associative Memories on Agricultural Context

Mario Aldape-Pérez, José Antonio Estrada Pavía,
Oscar Camacho-Nieto, and Joel Omar Juarez Gambino

Center for Computing Innovation and Technology Development. CIDETEC
National Polytechnic Institute, IPN
Mexico City, Mexico
maldape@ieee.org; jaestradap@gmail.com; oscarc@cic.ipn.mx;
omarjg82@gmail.com
URL: http://www.aldape.mx

Abstract. 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. In this paper we will compare Alpha-Beta associative memory against CHAT associative memory in order to find out which one is better for a real agronomic application focused on herbicides.

Keywords: Associative Memories, Alpha-Beta, CHAT, Herbicides, Pattern Recognition

1 Introduction

This paper focuses on classification of pesticides. Synthetic organic pesticides are used to control weeds, insects, and other organisms in a wide variety of agricultural and non-agricultural settings [1]. Particularly this paper is focused on herbicides that are used to control weeds.

Nowadays the agronomists have a lot of herbicides to use, as well as a lot of information related to product application. Each herbicide has its own characteristics, and more important has a set of weed and crops in where it can be used, herein lies a classification problem, a lot of information and difficult access to it. Since 1997 this problem has been taken into account.

On 1997 Artificial Neural Networks (ANN) have proven to be an effective alternative to deal with complex agricultural systems [2]. After this first approach, complex intelligent agronomic systems have been developed. In 2003 a herbicide application using neural networks and fuzzy logic was developed; results of this project illustrated the potential benefits of using image processing methods, ANNs and fuzzy logic to develop herbicide application maps for precision farming [3].

Later on 2004 a work was developed in order to investigate the potential of using neural-network classifiers to discriminate weed species in field crops. It has been found that the Back Propagation Neural Network (BPNN) classifier achieved classification accuracies of 96.7% which exceed discriminant classification procedures in relatively simple network topologies [4].

In 2005 an application of ANN was developed for prediction of pesticide occurrence in rural domestic wells with reduced information. With the available information, a four layer BPNN can be employed to detect pesticide occurrence in wells with 89% accuracy [1].

Some authors describe associative memories as weightless neural networks, however associative memories have intrinsic properties that make them suitable for a wide variety of problems [5],[6], [7], [8]. The fundamental purpose of an associative memory is to recall complete patterns from input patterns, which may be affeted with additive, subtractive or mixed noise [9]. Associative memories have two functional phases, the learning phase, and the recalling phase, in the learning phase the associative memory is built with a set of previously classified patterns one input pattern and its corresponding output pattern.

One of the most relevant associative memory work was developed by Hopfield in 1982 [10] his model demonstrates that the interaction of simple processing elements, similar to neurons, give place to the rise of collective computational properties, such as the stability of memories [9]. The associative memories models to be compared in this paper are named Alpha-Beta Associative Memory and CHAT Associative Memory.

2 Alpha-Beta Associative Memories

The purpose of any associative memory is to recall correct output patterns from input patterns, which may be altered with additive, subtractive or mixed noise. The concepts used in this section are presented in [9].

An associative memory M is a system that relates input patterns and output patterns as follows: $x \longrightarrow \boxed{\mathbf{M}} \longrightarrow y$ with x and y, respectively, the input and output pattern vectors. Each input vector forms an association with its corresponding output vector. For each k integer and positive, the corresponding association will be denoted as: (x^k, y^k). Associative memory M is represented by a matrix whose ij-th component is m_{ij} [11]. Memory 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: $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu\ \forall \mu \in \{1, 2, ..., p\}$ M is auto-associative, otherwise it is heteroassociative; in this case, it is possible to establish that $\exists \mu \in \{1, 2, ..., p\}$ for which $x^\mu \neq y^\mu$. If we consider the fundamental set of patterns $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ where n and m are the dimensions of the input patterns and output patterns, respectively, it is said that $x^\mu \in A^n$, $A = \{0, 1\}$ and $y^\mu \in A^m$. Then the j-th component of an input pattern is $x_j^\mu \in A$. Analogously, the j-th component of an output pattern is represented as $y_j^\mu \in A$. A distorted version of a pattern x^k to be recuperated will be denoted as $\widetilde{x}^k$. If when feeding an unknown input pattern x^ω with $\omega \in \{1, 2, ..., k, ..., p\}$

Table 1. Alfa and Beta Operators.

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

x	y	α(x,y)
0	0	1
0	1	0
1	0	2
1	1	1

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

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

to an associative memory M, it happens that the output corresponds exactly to the associated pattern y^{ω}, it is said that recuperation is perfect.

Alpha-Beta Associative Memories mathematical foundations are based on two binary operators: α and β. Alpha operator is used during the learning phase while Beta operator is used during the recalling phase. The mathematical properties within these operators, allow the $\alpha\beta$ associative memories to exhibit similar characteristics to the binary version of the morphological associative memories, in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall [12]. First, we define set $A = \{0, 1\}$ and set $B = \{0, 1, 2\}$, so α and β operators can be defined as in Table 1.

These two binary operators along with maximum ($\vee$) and minimum ($\wedge$) operators establish the mathematical tools around the Alpha-Beta model. The definitions of α and β exposed in Table 1, 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. According to the type of 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 of type MAX will be obtained, denoted as M; analogously, if minimum operator ($\wedge$) is used, Alpha-Beta Associative Memory of type min will be obtained, denoted as W [9]. In any case, the fundamental input and output patterns are represented as follows:

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

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

α max Operation: $P_{mxr} \nabla_{\alpha} Q_{rxn} = \left[f_{ij}^{\alpha}\right]_{mxn}$, where $f_{ij}^{\alpha} = \vee_{k=1}^{r} \alpha(p_{ik}, q_{kj})$
β max Operation: $P_{mxr} \nabla_{\beta} Q_{rxn} = \left[f_{ij}^{\beta}\right]_{mxn}$, where $f_{ij}^{\beta} = \vee_{k=1}^{r} \beta(p_{ik}, q_{kj})$
α min Operation: $P_{mxr} \Delta_{\alpha} Q_{rxn} = \left[f_{ij}^{\alpha}\right]_{mxn}$, where $f_{ij}^{\alpha} = \wedge_{k=1}^{r} \alpha(p_{ik}, q_{kj})$
β min Operation: $P_{mxr} \Delta_{\beta} Q_{rxn} = \left[f_{ij}^{\beta}\right]_{mxn}$, 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:

$$y\nabla_\alpha x^t = y \oplus x^t = y\Delta_\alpha x^t.$$

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

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

2.1 Learning Phase

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

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

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

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

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

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

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

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

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

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

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

2.2 Recalling Phase

Find the adequate operators and sufficient conditions to obtain the fundamental output pattern y^μ, when either the memory M or the memory W is operated with the fundamental input pattern x^μ.

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

Alpha-Beta Associative Memory of type MAX:

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

Alpha-Beta Associative Memory of type min:

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

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

3 CHAT Associative Memory

Associative Memory CHAT is based on the combination of two associative memories, the Lernmatrix from Steinbuch[13] and Linear Associator from Kohonen[11]. By combining these two associative memories the Associative Memory CHAT can accept real values in the components of the input vectors and these vectors are not required to be orthonormal, to obtain those new properties the learning phase from Linear Associator and the recalling phase from Lernmatrix were combined [14].

The algorithm for memory CHAT is as follows:

1. Lets define a fundamental set of input patterns of dimension n with real values on their components (as in the Linear Associator), this pattern are organized in m different classes.
2. For each input pattern in the class k define a vector made of zeros except the $k - th$ coordinate, where the value is one (as in the Lernmatrix).
3. Calculate the medium vector from the set of sample patterns with the expression.

$$\bar{x} = \frac{1}{p} \sum_{\mu=1}^{p} x^\mu$$

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

4. Take the components from the medium vector as the center of a new set of coordinate axes.
5. Move all the patterns from the fundamental set.

$$x^{\mu'} = x^{\mu} - \bar{x}$$

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

6. Apply the learning phase as described on [14]. This phase is similar to the Linear Associator learning phase.

The learning phase involves two stages:

Stage 1 Calculate each one of the p associations (x^{μ}, y^{μ}) in order to find the matrix $y^{\mu} \cdot (x^{\mu})^t$ with dimentions $m \times n$.

$$y^{\mu} \cdot (x^{\mu})^t = \begin{pmatrix} y_1^{\mu} \\ y_2^{\mu} \\ \vdots \\ y_m^{\mu} \end{pmatrix} \cdot (x_1^{\mu}, x_2^{\mu}, ..., x_n^{\mu})$$

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

Stage 2 Add the p matrices to obtain the memory M

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

so that the ij-th component of memory M is expressed as follows:

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

7. Move all the patterns to be classified to the new axis

$$x^{\mu'} = x^{\mu} - \bar{x}$$

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

8. Apply the recalling phase as in [14]. This phase is similarly to the Lernmatrix recalling phase.

The recalling phase is devoted to find the class that belongs to an input vector $x^\omega \in A^n$ given. Find the class means to obtain the components from the vector $y^\omega \in A^p$ that belongs to the pattern x^ω ; with the construction method of the vectors y^μ, class label should be obtained without ambiguity.

The i-th coordinate y_i^ω from the vector with the class $y^\omega \in A^p$ is obtained as shown in the next expression, where $\bigvee$ is the maximum operator

$$y_i^\omega = \begin{cases} 0 & if \quad \sum_{j=1}^{n} m_{ij} \cdot x_j^\omega = \bigvee_{h=1}^{p} \left[\sum_{j=1}^{n} m_{hj} \cdot x_j^\omega \right] \\ 1 & \text{other case} \end{cases}$$

This memory is being improved nowadays by combining CHAT memory with different coding techniques, like in [15]. This new form of CHAT memory can improve the classification accuracy.

4 Herbicide Dataset

The dataset used along experimental phase is an extraction from the Dictionary of Agrochemical Specialities (DEAQ). The DEAQ can be consulted online in [16], the main characteristics of 40 herbicides were distributed in 3473 instances. The features are: toxicity, action method, active ingredient, incompatibility, reentry period, counter indications and first aid.

In the dataset the patterns are structured as follows:

{"weed", "crop", "toxicity", “reentry period ", "herbicide"}

Where "weed", "crop", "toxicity" and "reentry period" are the principal attributes to be analyzed and "herbicide" is the most suitable option. The dataset contains 68 weeds, 50 crops, 2 kinds of toxicity, and 6 different reentry periods.

5 Experimental Phase

The fundamental set made from 40 herbicides distributed in 3473 instances was taken on all the experiments. Two experiments were performed with the associative models. The experimental procedure is as follows: first a k-fold cross validation classification was performed with CHAT Associative Memory in order to test the recovery rate over the fundamental set, then the learning model was changed to Alpha-Beta Associative Memory where recovery rate was evaluated.

In order to determine the classification performance of other existing models in the literature, a set of experiments were performed with "WEKA 3: Data Mining Software in Java" [17]. WEKA is an open source application under the GNU General Public License, free available on internet. On all the tests with WEKA the same conditions and validation schemes were aplied. The algorithms were trained with all the fundamental set and then the fundamental set was recovered. In Table 2 all the resullts are compared. In the following paragraph the experiments with associative models are explained.

The first approach was to use CHAT Associative Memory. This associative model is known to be a good choice to classify patterns in bi-class problems, this has been tested by classifying each of the products into one of the two possible classes: "is the product" or "is not the product", this experimental approach was performed using a different memory for each product. CHAT Associative Memory was created only with all the training patterns from a specific product, for the "not product" examples, the pattern that doesn't have the product was taken, until it complete the same number of correct patterns. This experiment gave good results, with a k-fold cross validation with k=10, classification efficiency from all the products was 80.27%.

The next model was a cascade model made with a set of CHAT Associative Memories, each CHAT Associative Memory was trained with a set of products and "not products". In this approach the memories were organized from most efficient to the less efficient, this in order to avoid error propagation in the model, then CHAT Associative Memories were trained with the crop feature, so now there was a set of memories that only would be used with each different crop, using this model. This approach was performed in order to know how much the new model can recover from the learned phase, the result was a 21.01% . This implies that this model can recover from its fundamental set, however as the goal was to select a product, with this result this cannot be done in a competitive way even if the user gave a pattern contained in the fundamental set.

This forced us to seek for a more efficient solution, which was to change the classification mode to Alpha-Beta Associative Memories. First all the patterns were codified with The Johnson-Möbius code to improve the efficiency from Alpha-Beta Associative Memories[5], then the autoassociative Alpha-Beta Asociative Memories were selected, due to the fact that autoassociative Alpha-Beta Asociative Memories can recover entirely the fundamental set learned, classification efficiency reached 100% . This implies that the 3473 instances were correctly classified.

6 Experimental Results

Although WEKA 3: Data Mining Software in Java[17] has more than seventy well known algorithms implemented, only the two best-performing algorithms from each category were considered for comparison purposes. 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, Lazy based classifiers and Decision Trees classifiers.

Decision trees algorithms can classify the fundamental set with an averange of 71.14%, the bayesian algorithms could classify the fundamental set with a BayesNet but not with a NaiveBayes, the best category for classify the fundamental set was lazy, with an average of 72.27%, the function based classifiers are

Table 2. Eficiency recovering fundamental set.

Algorithm	Instances correctly classified	Percentage
Trees		
RandomForest	2505	72.13%
NBTree	2437	70.16%
Bayesian		
BayesNet	2484	71.52%
NaiveBayes	1637	47.13%
Functions		
MultilayerPercerpron	2026	58.33%
Logistic	1897	54.62%
Lazy		
IB1	2510	72.27%
IBK	2510	72.27%
Meta		
Bagging	2444	70.37%
AttributeSelectedClassifier	2361	67.98%
Associative Memories		
Alpha-Beta	**3473**	**100%**
CHAT	730	21.01%

not good enough to classify the fundamental set with an averange of 56.47%. On the associative models CHAT Associative Memory was not the best solution even though it was mixed in different models trying to improve the classification rate, but best of all algorithms was the Alpha-Beta Associative Memories with a 100% of correct classification from the fundamental set. This imply that everything reminds learned on the Alpha-Beta Associative Memories.

7 Conclusions and Ongoing Research

Alpha-Beta Associative Memories can recall the entire fundamental set, this give us a big opportunity to develop a wide range of applications. In this paper the Alpha-Beta Associative Memory and CHAT Associative Memory were compared, in this particular case, an agronomic application was created to make recommendations about a herbicide. The best memory to use in this agronomic application was the Alpha-Beta memory, due its characteristics. On each intelligent application developed an important point to consider is the classification performance. Finally this paper shows that a problem in an agricultural context can be solved using a specific characteristic of a classification technique, in this case Alpha-Beta Associative memories.

Currently, we are investigating how to make better associative memories using feature selection in order to include them in more fields of science.

References

1. Sahoo, G., Ray, C., Wade, H.: Pesticide prediction in ground water in north carolina domestic wells using artificial neural networks. Ecological Modelling **183** (2005) 29 – 46
2. Hashimoto, Y.: Applications of artificial neural networks and genetic algorithms to agricultural systems. Computers and Electronics in Agriculture **18** (1997) 71 – 72 Applications of Artificial Neural Networks and Genetic Algorithms to Agricultural Systems.
3. Yang, C.C., Prasher, S.O., Landry, J.A., Ramaswamy, H.S.: Development of a herbicide application map using artificial neural networks and fuzzy logic. Agricultural Systems **76** (2003) 561 – 574
4. Burks, T., Shearer, S., Heath, J., Donohue, K.: Evaluation of neural-network classifiers for weed species discrimination. Biosystems Engineering **91** (2005) 293 – 304
5. Cornelio Yañez Edgardo, Felipe-Riveron, I.L.Y., Flores-Carapia, R.: A novel approach to automatic color matching. In Martinez-Trinidad, J., ed.: CIARP 2006. LNCS (2006) 529–538
6. Yáñez-Márquez, C., Cruz-Meza, M.E., Sánchez-Garfias, F.A., López-Yáñez, I.: Using alpha-beta associative memories to learn and recall RGB images. In: Advances in Neural Networks - ISNN 2007, 4th International Symposium on Neural Networks, ISNN 2007, Nanjing, China, June 3-7, 2007, Proceedings, Part III. (2007) 828–833
7. Román-Godínez, I., López-Yáñez, I., Yáñez-Márquez, C.: Classifying patterns in bioinformatics databases by using alpha-beta associative memories. In: Biomedical Data and Applications. Springer-Verlag Berlin Heidelberg (2009) 187–210
8. Aldape-Pérez, M., Román-Godínez, I., Camacho-Nieto, O.: Thresholded learning matrix for efficient pattern recalling. In: CIARP '08: Proceedings of the 13th Iberoamerican congress on Pattern Recognition, Berlin, Heidelberg, Springer-Verlag (2008) 445–452
9. Acevedo-Mosqueda, Yañez-Marquez, L.Y.: Alpha. beta bidirectional associative memories: theory and applications. Neural Processing Letters (2007)
10. JJ, H.: Neural networks and physical systems with emergent collective computational abilities. Proc Nat AcadSci (1982)
11. Kohonen, T.: Correlation matrix memories. IEEE Transactions on Computers (1972)
12. Acevedo-Mosqueda, Yañez-Marquez, L.Y.: A new model of bam: Alpha-beta bidirectional associative memories. Lecture Notes in Computer Science (LNCS) (2006)
13. Steinbuch, K., Frank, H.: Nichtdigitale lernmatrizen als perzeptoren. Biological Cybernetics **1** (1961) 117–124
14. Montero, R.S.: Clasificador hibrido de patrones basado en la lernmatrix de steinbuch y el linear associator de anderson-kohonen. Master's thesis, Instituto Politecnico Nacional Centro de Investigacion en Computacion (2003)
15. Uriarte-Arcia AV, López-Yáñez I, Y.M.C.: One-hot vector hybrid associative classifier for medical data classification. PLoS ONE **9** (2014) e95715
16. PLM: Diccionario de especialidades agroquimicas [online]. http://www.elcamporadio.com/source/ (2014)
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter **11** (2009) 10–18