Feature Selection on Associative Models using Single Board Computer Paradigm

R. Ramírez-Rubio, E. Solórzano-Alor, M. Aldape-Pérez and C. Yáñez-Márquez

Instituto Politécnico Nacional, IPN
Centro de Innovación y Desarrollo Tecnológico en Cómputo, CIDETEC
México D.F., México
rogelioramirezr@hotmail.com; eduardosolorzano22@hotmail.com;
maldape@gmail.com; coryanez@gmail.com

Abstract. In this paper feature selection and associative models are addressed. Classification accuracy is evaluated using six medical datasets, widely used in Machine Learning. In order to obtain the optimal subset of features, an associative model was implemented on a Single Board Computer (SBC).

Keywords: Feature Selection, Single Board Computer, Associative Model.

1 Introduction

Most of the times the initial set of features consists of a large number of potential attributes that constitute an obstacle not only to the accuracy but to the efficiency of algorithms. In high dimensional spaces features often tend to be correlated, in this way a successful subset of features provided to a classifier can increase its accuracy, save computation time, and simplify its results [9]. The design of pattern classifiers has at least three goals: (1) to reduce the cost of extracting features, (2) to improve classification accuracy, and (3) to improve performance reliability [8].

This paper presents *Hybrid Classification and Masking* algorithm (HCM) implemented on a *Single Board Computer* (SBC). The main idea is to compare the performance of HCM with other classification algorithms to determine a level of competitiveness for different applications. The main idea is that a strong classifier could be obtained when a hybrid associative memory is used during the learning phase thus additional advantages appear. Since one of the associative memories properties is to establish links between patterns and classes to which they belong on a one-shot basis, iterative complexity is eliminated. This means that one and only one classifier is computed during the whole classification and masking process.

Another clear advantage during the second phase (recalling) arises when any kind of prior knowledge is not needed in the sense that no distributional or functional assumptions are considered. Moreover the optimal mask search algorithm is applied only to those patterns that were previously considered during the first phase (learning), which means that no additional patterns are required to increase classifier accuracy.

2 Main Concepts

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} [10]. 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 $y_j^\mu \in A$.

2.1 Lernmatrix

Lernmatrix is a heteroassociative memory that can easily work as a binary pattern classifier if output patterns are appropriately chosen [13]. It accepts binary patterns suchlike $\mathbf{x}^\mu \in A^n$, $A = \{0, 1\}$ as input and returns binary patterns suchlike $\mathbf{y}^\mu \in A^m$ as output; it is worth pointing out that there are m different classes, each one coded by a simple rule: class $k \in \{1, 2, ..., m\}$ will be represented by a column vector which components will be assigned by $y_k^\mu = 1$, so $y_j^\mu = 0$ for $j = 1, 2..., k-1, k+1, ...m$.

The following matrix will keep the pattern association values after the Learning Phase for the Steinbuch's *Lernmatrix* is done:

$$\begin{array}{c|c|c|c|c|c|c|}
 & x_1^\mu & x_2^\mu & \cdots & x_j^\mu & \cdots & x_n^\mu \\
\hline
y_1^\mu & m_{11} & m_{12} & \cdots & m_{1j} & \cdots & m_{1n} \\
\hline
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
\hline
y_i^\mu & m_{i1} & m_{i2} & \cdots & m_{ij} & \cdots & m_{in} \\
\hline
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
\hline
y_m^\mu & m_{m1} & m_{m2} & \cdots & m_{mj} & \cdots & m_{mn} \\
\hline
\end{array} \tag{1}$$

Each one of the m_{ij} components of $\mathbf{M}$ is initialized with zero and will be modified by the following rule: $m_{ij} = m_{ij} + \Delta m_{ij}$ where:

$$\Delta m_{ij} = \begin{cases} +\varepsilon & if \quad y_i^\mu \quad = \quad 1 \quad = \quad x_j^\mu \\ -\varepsilon & if \quad y_i^\mu \quad = \quad 1 \quad and \quad x_j^\mu \quad = \quad 0 \\ 0 & otherwise \end{cases} \tag{2}$$

and ε a positive constant, previously chosen.

The Recalling Phase for the Steinbuch's *Lernmatrix* consists of finding the class which an input pattern $\mathbf{x}^\omega \in A^n$ belongs to. Finding the class means getting $\mathbf{y}^\omega \in A^m$ that corresponds to $\mathbf{x}^\omega$; accordingly to the construction method of all $\mathbf{y}^\mu$, the class should be obtained without ambiguity. The *i*-th component of y_i^ω is obtained according to the following rule, where $\vee$ is the *maximum* operator:

$$y_i^\omega = \begin{cases} 1 & \text{if} \quad \sum_{j=1}^{n} m_{ij}.x_j^\omega = \bigvee_{h=1}^{m} \left[\sum_{j=1}^{n} m_{hj}.x_j^\omega \right] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2.2 Linear Associator

Lets consider the fundamental set as $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1, 2, ..., p\}$ with

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

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 $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$

$$\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, ..., x_n^\mu) = \begin{pmatrix} y_1^\mu x_1^\mu & \cdots & y_1^\mu x_j^\mu & \cdots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & \cdots & y_2^\mu x_j^\mu & \cdots & y_2^\mu x_n^\mu \\ \vdots & \vdots & \vdots & & \vdots \\ y_i^\mu x_1^\mu & \cdots & y_i^\mu x_j^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & \vdots & & \vdots \\ y_m^\mu x_1^\mu & \cdots & y_m^\mu x_j^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix} \tag{4}$$

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

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \tag{5}$$

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

$$m_{ij} = \sum_{\mu=1}^{p} y_i^{\mu} x_j^{\mu} \tag{6}$$

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

$$\mathbf{M} \cdot \mathbf{x}^{\omega} = \left[\sum_{\mu=1}^{p} \mathbf{y}^{\mu} \cdot (\mathbf{x}^{\mu})^{t} \right] \cdot \mathbf{x}^{\omega} \tag{7}$$

Lets expand 7, we obtain:

$$\mathbf{M} \cdot \mathbf{x}^{\omega} = \mathbf{y}^{\omega} \cdot \left[(\mathbf{x}^{\omega})^{t} \cdot \mathbf{x}^{\omega} \right] + \sum_{\mu \neq \omega} \mathbf{y}^{\mu} \cdot \left[(\mathbf{x}^{\mu})^{t} \cdot \mathbf{x}^{\omega} \right] \tag{8}$$

Expression 8 lets us know about which restrictions have to be observed thus perfect recalling is achieved. These restrictions are:

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

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

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

$$(\mathbf{x}^{\mu})^{t} \cdot \mathbf{x}^{\omega} = \begin{cases} 1 & \text{if} \quad \mu = \omega \\ 0 & \text{if} \quad \mu \neq \omega \end{cases} \tag{9}$$

If condition 9 is met, then a perfect recalling is expected. So 8 is expressed as:

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

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

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

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

This term is known as *cross-talk*, it represents some kind of noise that comes from input patterns interaction. As a consequence perfect recalling is not achieved, except in those cases where the number of stored patterns is rather small compared to n . Some researchers' results confirm that this number should be between $0.1n$ and $0.2n$ [2],[5], [11].

2.3 Hybrid Associative Classifier with Translation

Hybrid Associative Classifier with Translation (CHAT for its acronym in Spanish) combines *Linear Associator* learning phase and *Lernmatrix* recalling phase [6]. The algorithm of the *Hybrid Associative Classifier with Translation* [12] model is as follows:

1. Let n be the dimension of each input pattern in the fundamental set, grouped in m different classes.
2. Each one of the input patterns belongs to a k class, $k \in \{1, 2, ..., m\}$, represented by a column vector which components will be assigned by $y_k^\mu = 1$, so $y_j^\mu = 0$ for $j = 1, 2..., k-1, k+1, ...m$.
3. The learning phase is carried out as a *Linear Associator* , according to expression 4, 5 and 6.
4. The recalling phase is carried out as a *Lernmatrix* according to expression 3.

2.4 Hybrid Classification and Masking Approach

Hybrid Classification and Masking technique (HCM) is presented as a new feature selection approach to provide a mask that identifies the optimal subset of features without having to compute a new classifier at each step. This method allows us to identify irrelevant or redundant features for classification purposes [1]. In order to explain how optimal mask is found, some definitions are required.

Definition 1. *Let* $\mathbf{f}$ *be the number of features from the original set of data.*

Definition 2. *Let* $\mathbf{r}$ *be an index where* $r \in \{1, 2, ..., (2^f - 1)\}$

Definition 3. *Let* $\mathbf{e}^r$ *be a masking vector of size* n *represented as:*

$$\mathbf{e}^r = \begin{pmatrix} e_1^r \\ e_2^r \\ \vdots \\ e_n^r \end{pmatrix} \in B^n \tag{10}$$

where $B = \{0, 1\}$

Definition 4. *Let* $\dashv$ *be a new operation called* $IntToVector$ *which takes* $r \in \{1, 2, ..., (2^f - 1)\}$ *and returns a column vector* $\mathbf{e}^r$ *with* r *value expressed in its binary form. From a register transfer level perspective (RTL) this can be expressed as* $bin(r) \rightarrow [\mathbf{e}^r]$ *. For example: If* $r = 11$ *then* $\dashv \mathbf{e}^r$ *returns a column vector with* r *value in its binary form so the obtained vector is:*

$$\mathbf{e}^{11} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

where e_n^r is the Least Significant Bit (LSB)

Definition 5. *Let $\|$ be a new operation called $MagVector$ which takes a column vector $\mathbf{e}^r$ of size n and returns an integer and positive value according to the following rule:*

$$\| \, \mathbf{e}^r = \sum_{j=1}^{n} \left(e_j^r \wedge 1 \right) \tag{11}$$

Where $\wedge$ is the logical AND operator.

Another relevant thing to mention is that the $Recalling\ Phase$ is dramatically different from the previous models; it is carried out by the following rule:

$$y_i^\mu = \begin{cases} 1 & \text{if} \quad \sum_{j=1}^{n} m_{ij} \cdot \left(x_j^\mu . e_j^r \right) = \bigvee_{h=1}^{m} \left[\sum_{j=1}^{n} m_{hj} \cdot \left(x_j^\mu . e_j^r \right) \right] \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\text{where } \mu \in \{1, 2, ..., p\} \text{ and } r \in \{1, 2, ..., (2^f - 1)\}$$

It is said that the $Recalling\ Phase$ is dramatically different from the previous models because a masking vector $\mathbf{e}^r$ of size n masks each input vector x^μ of size n. This is where the masking technique comes into view. Using the previous definitions and the clear advantages that inherit from the $Hybrid\ Associative\ Classifier$ model, it is possible to enunciate The HCM algorithm.

1. Create a classifier using (4), (5) and (6).
2. Use the $IntToVector$ operator to get the r-th masking vector as in (10)
3. The recalling phase is carried out according to expression (12) so an r-th accuracy parameter is obtained
4. Store both parameters (the r-th accuracy parameter and the r-th masking vector) so feature selection can be evaluated in step (6)
5. Compare the r-th accuracy parameter with the $(r-1)$-th accuracy parameter. The best accuracy value is stored thus accuracy improvements are achieved with each iteration
6. The same applies to the r-th masking vector. Feature selection can be evaluated using expression (11). So the smaller this number is, a better mask is obtained.
7. The new subset of features is obtained by a mask value represented by a column vector, where accuracy and feature selection are optimal

2.5 Single Board Computer

The *SBC paradigm* is very effective for specific applications, for which it was designed [4], this paradigm can be used as a tool for tasks requiring precision control and data acquisition, which is consider scientific applications [7] and industrial applications such as remote monitoring and gas tank filling control [3]. The Parallella board is a high performance computing device, which can be used as a standalone computer (SBC) or as an embedded system. The Parallella platform includes a low-power consumption pro dual core ARM A9 processor , which is able to work with different Linux distributions, it makes it attractive by the versatility of having many work environments and give some facilities to the end users. Also this board contain a 1 GB DDR3 RAM and a slot for micro SD memory. One of the applications that can be given to Parallella board is image processing and analysis, a clear example is face detection, this issue has been extensively studied with low-cost computing and innovative algorithms [16].

3 Datasets

The following datasets were used because of their confidence and wide study in this area. These datasets have been used to probe and validate new classifiers as well as classical classification techniques.

Breast Cancer: This database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg and was donated by Olvi Mangasarian. This dataset contains periodical samples of clinical cases. Breast cancer dataset consists of 699 instances belonging to two different classes (458 benign cases, 241 malign cases). Each instance consists of 10 attributes, including the class attribute.

Heart Disease: This database comes from the Cleveland Clinic Foundation and was supplied by Robert Detrano, M.D., Ph.D. of the V.A. Medical Center, Long Beach, CA. The purpose of the dataset is to predict the presence or absence of heart disease given the results of various medical tests carried out on a patient. This dataset consists of 270 instances belonging to two different classes: presence and absence (of heart-disease). Each instance consists of 14 attributes, including the class attribute.

Hepatitis Disease: This dataset was donated by the Jozef Stefan Institute, former Yugoslavia, now Slovenia. The purpose of the dataset is to predict the presence or absence of hepatitis disease in a patient. Hepatitis disease dataset consists of 155 instances belonging to two different classes (32 die cases, 123 live cases). Each instance consists of 20 attributes, 13 binary, 6 attributes with discrete values and a class label.

Liver Disorders: This database was created by BUPA Medical Research Ltd and was donated by Richard S. Forsyth. This dataset contains cases from a study that was conducted on liver disorders that might arise from excessive alcohol consumption. Liver disorders dataset consists of 345 instances belonging to two different classes. Each instance consists of 7 attributes, including the class attribute.

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

Pima Indians Diabetes: This database was originally owned by the National Institute of Diabetes and Digestive and Kidney Diseases, U.S. This dataset contains cases from a study that was conducted on female patients at least 21 years old of Pima Indian heritage. This dataset consists of 768 instances belonging to two different classes (500 the patient tested positive for diabetes cases, 268 the patient tested negative for diabetes cases). Each instance consists of 9 attributes, including the class attribute.

4 Experimental Phase

The experimental phase was conducted with twenty - one algorithms all of them are different pattern classification methods, executed in WEKA enviroment [15], except for the proposed algorithm. Six medical datasets obtained from the UCI: Machine Learning Repository [14] were used in order to test the efficiency (the percentage of correct classification) and average of each one of them. In addition, algorithms proposed were compared, to view the execution times as well as the improved of efficiency, the algorithms were executed in a Parallella Board.

Table 1 shows the classification accuracy of the algorithms.

Table 3 shows the execution time comparison of CHAT before and after feature selection.

As can be seen in Table 4, classification accuracy and execution time were improved in CHAT, after the feature selection.

Table 1: Efficiency (%) of classification of each algorithm

Dataset Algorithm	Breast	Heart	Hepatitis	Liver	Parkinson	Pima	Average
AdaBoostM1	95.60	81.85	60.00	63.73	85.12	75.65	76.99
Bagging	95.90	81.11	68.38	70.43	87.69	74.73	79.71
BayesNet	97.36	**83.70**	70.96	63.18	80.00	75.78	78.50
Dagging	96.92	82.96	63.87	57.68	85.12	74.21	76.79
DecisionTable	95.75	82.96	**73.54**	63.18	83.58	74.86	78.98
DTNB	97.07	80.74	69.67	63.18	85.12	74.86	78.44
FT	97.21	80.37	69.03	**71.59**	84.61	76.69	79.92
LMT	96.19	**83.70**	66.45	69.85	86.15	77.08	79.90
Logistic	96.77	82.96	69.67	68.40	86.66	**77.73**	**80.36**
MultiClassClassifier	96.77	82.96	69.67	68.40	86.66	**77.73**	**80.36**
NaiveBayes	96.19	82.96	71.61	55.94	69.23	75.65	75.26
NaiveBayesSimple	96.33	**83.70**	70.96	55.36	69.23	75.26	75.14
NveBayesUpdateable	96.19	82.96	71.61	55.94	69.23	75.65	75.26
RandomCommittee	96.63	82.22	62.58	68.11	**90.76**	75.39	79.28
RandomForest	96.48	82.59	60.64	67.82	**90.76**	74.60	78.81
RandomSubSpace	95.90	81.11	67.09	68.69	88.7	73.30	79.13
RBFNetwork	96.19	81.85	71.61	64.63	84.10	73.69	78.68
RotationForest	**97.51**	79.62	64.51	70.72	90.25	76.56	79.86
SimpleLogistic	96.48	**83.70**	65.16	69.27	84.61	77.34	79.43
SMO	97.07	82.96	69.67	57.97	87.17	76.82	78.61
CHAT	**97.51**	64.07	66.45	55.36	68.72	61.45	68.93

Table 2: Classification efficiency (%) of CHAT and CHAT with Feature Selection

Dataset Algorithm	Breast	Heart	Hepatitis	Liver	Parkinson	Pima	Average
CHAT	97.51	64.07	66.45	55.36	68.72	61.45	68.93
CHAT(FS)	97.80	83.70	85.16	65.50	70.76	70.96	**78.98**

Table 3: Classification Time (Seconds) of the complete dataset using CHAT and CHAT with Feature Selection

Dataset Algorithm	Breast	Heart	Hepatitis	Liver	Parkinson	Pima	Average
CHAT	0.006673	0.003724	0.002654	0.002747	0.003317	0.007171	0.004381
CHAT(FS)	0.005364	0.001954	0.000793	0.002120	0.001641	0.005143	**0.002835**

Table 4: Improved classification accuracy and time saved (%) by using feature selection in CHAT compared with original features

Dataset / Algorithm	Breast	Heart	Hepatitis	Liver	Parkinson	Pima	Average
Time	19.61	47.52	70.12	22.82	50.52	28.28	**39.81**
Classification	0.2928	19.62	18.70	10.14	2.05	9.50	**10.05**

5 Results Analysis and Conclusion

As shown in Table 1 the associative model CHAT shows competitive an performance compared with other classification algorithms, which makes it an option for classification. When added the feature selection technique, it is observed that the CHAT improved considerably in classification as shown in Table 2, also the classification obtained in less time, as it is shown in Table 3. Table 4 shows CHAT with feature selection, and improve the CHAT efficiency in time and classification accuracy.

Now, according with the previous work we can conclude that as shown in Table 1, CHAT proves competitive performance against other classification algorithms present in the state of the art either individually or averaged.

With respect to Table 2, it can be concluded that classification efficiency has undergone CHAT feature selection to the data set used, considerably improves from 68.93% to 78.98% on average.

In Table 3, the classification time of complete sets are shown, which in principle are low. It is worth noting that classification time for each instance not only is good but in average is acceptable considering that datasets that were used in the experimental phase range from nine to twenty-two features (excluding the class label).

Finally in Table 4, the time savings are observed for each database and the average of them is 39.81%. Likewise classification improvement of each database is shown and the average of them is 10.05%.

It is noteworthy that feature selection has its respective cost, but it must be paid only once. Moreover this is compensated with subsequent classifications as more reliable and less time consuming results are obtained. These advantages will be used in each and every one of the instances to be classified in the future.

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

References

1. M. Aldape-Perez, C. Yanez-Marquez, and L.O. Lopez Leyva. Feature selection using a hybrid associative classifier with masking techniques. In *Artificial Intelligence, 2006. MICAI '06. Fifth Mexican International Conference on*, pages 151–160, Nov 2006.
2. James A. Anderson and Edward Rosenfeld, editors. *Neurocomputing: Foundations of Research.* MIT Press, Cambridge, MA, USA, 1988.
3. L. Benussi, M. Bertani, S. Bianco, F.L. Fabbri, P. Gianotti, V. Lucherini, E. Pace, N. Qaiser, and S. Sarwar. Design and implementation of an automatic gas-cylinder inversion system based on an embedded computer. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 461(13):98 – 99, 2001. 8th Pisa Meeting on Advanced Detectors.
4. Paul A. Braun, Randy H. Moss, and Paul D. Stigall. The single board computer: A practical alternative to a custom designed microsystem. *Computers & Electrical Engineering*, 12(34):109 – 118, 1986.
5. Mohamad H. Hassoun. *Fundamentals of Artificial Neural Networks.* MIT Press, Cambridge, MA, USA, 1st edition, 1995.
6. G. Dreyfus I.G.L. Personnaz. Information storage and retrieval in spin glass like neural networks. *Journal of Physical Letters*, 46:359 – 365, 1985.
7. Haidong Kim and Matthew J. Zabik. Design and evaluation of a versatile single board computer for embedded applications in scientific instrumentation. *Computers & Chemistry*, 16(3):261 – 263, 1992.
8. Mineichi Kudo and Jack Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25 – 41, 2000.
9. Mark Last, Abraham Kandel, and Oded Maimon. Information-theoretic algorithm for feature selection. *Pattern Recognition Letters*, 22(67):799 – 811, 2001.
10. Gunther Palm, Friedhelm Schwenker, Friedrich T Sommer, and Alfred Strey. Neural associative memories. *Associative processing and processors*, pages 307–326, 1997.
11. G.X. Ritter, P. Sussner, and J.L. Diza-de Leon. Morphological associative memories. *Neural Networks, IEEE Transactions on*, 9(2):281–293, Mar 1998.
12. R. Santiago-Montero. Hybrid Associative Classifier Based on Lernmatrix and Linear Associator (In Spanish. Master's thesis, Center for Computing Research, México, 2003.
13. K. Steinbuch. Die lernmatrix. *Kybernetik*, 1(1):36 – 45, 1961.
14. UCI. Machine learning repository. Accessed on 16-05-2015 a url-http://archive.ics.uci.edu/ml/, 2007.
15. Brian Hardyment Liang Liu Vladimir Petko Duncan Willcock. Machine learning group at the university of waikato. Accessed on 16-05-2015 a url-http://www.cs.waikato.ac.nz/ml/weka/downloading.html, 1993.
16. Ming Yang, James Crenshaw, Bruce Augustine, Russell Mareachen, and Ying Wu. Adaboost-based face detection for embedded systems. *Computer Vision and Image Understanding*, 114(11):1116 – 1125, 2010. Special issue on Embedded Vision.