AN ENSEMBLE AVERAGE CLASSIFIER FOR PATTERN RECOGNITION MACHINES

S.H. EL-KOMYATY, F.W. ZAKI, A.I. ABU EL-ZETTAR, and Y.M.I. [illegible]

ABSTRACT:

The ensemble average classification method introduced here is a new nonparametric classification procedure. In this method, ensemble average of training pattern vectors in each class is stored in a computer memory. Classification of an unknown pattern vector depends primarily on the difference between the stored ensemble average vectors and the unknown pattern vector. Performance of the new method in comparison with Bayes (optimal) and perceptron classifiers has been studied through a series of computer experiments. The results obtained showed that the new method provides as higher classification rate as the Bayes classifier. However, it requires less computation complexity and higher storage memory than both Bayesian and perceptron classifiers.

1-INTRODUCTION:

The problem of classification in pattern recognition systems arises when a pattern vector (consisting of a number of measurements taken from an unknown pattern) is required to be classified into one of several classes, on the basis of the vector measurements, such that the misclassification error is minimum. Several classification procedure have been reported in literature [1, 2, 3]. These may be categorised into two main types, parametric and nonparametric classifiers.

In parametric classifiers, it is assumed that there are a finite number of classes from which the pattern vector may have come and each class is characterised by a probability distribution of the measurements. For this case the problem may be considered as a statistical decision problem, that is having a number of hypotheses with a given distribution of the measurements, one of these hypotheses must be accepted and the others are rejected.

Parametric classifiers (always referred to as Bayes classifiers) lead to a minimal probability of classification error. However, they can be applied only when the probability distributions are known, a quite unrealistic situation in practice. In most of the pattern recognition problems, the available information consists only of a collection of correctly classified patterns which can be used to train the classifier.

** Dept. of Computer and Control Eng., Faculty of Engineering, El-Mansoura Univ.

* Dept. of Communication Eng., Faculty of Engineering, El-Mansoura University.

In nonparametric classifiers, the apriori knowledge of probability distributions and probability densities are unnecessary. Instead, a training set of pattern vectors belonging to the different classes are used for machine learning. The experience gained from this training set is stored to make use of it in classifying any unknown pattern vector. Among of these classifiers are the nearest neighbour classifier [4,6] and the linear discriminant functions classifier (perceptron)[5].

In this paper, a new nonparametric classifier called ensemble average classifier is introduced. This technique does not require apriori probability information. It requires only the calculation of K ensemble average pattern vectors, each of N dimensional vector, where K is the number of classes and N is the dimension of the pattern space. Classification of an unknown pattern, as will be seen later, depends on the difference between the ensemble average pattern vector and the unknown pattern vector.

II-BAYES CLASSIFIER

Let $\underline{X}$ be an N-dimensional pattern vector in the N-dimensional Euclidian space Ω_x. For a set of K classes $C= C_i$, $i=[1,2,.....,K]$, let P_k the prior probability that $\underline{X}$ belongs to C_k , and $f_k(\underline{X})=f(\underline{X}/C_k)$ be the conditional density function. Assuming equal costs of misclassification for all classes, the Bayes classifier[2,3] that minimizes the error probability will divide Ω_x into K disjoint regions with each region Ω_j consisting of values of $\underline{X}$ such that

$$P_j f_j(\underline{X}) \geq P_k f_k(\underline{X}) \quad \text{for all } k \neq j \qquad (1)$$

Assume that each $f_k(\underline{X})$ is an N-dimensional Gaussian density function with mean vector $\underline{m}_k$ and Covariance matrix R_k i.e,

$$f_k(\underline{X}) = (2\pi)^{-0.5N} |R_k|^{-0.5} \exp.\left[-0.5(\underline{X}-\underline{m}_k)^T R_k^{-1} (\underline{X}-\underline{m}_k)\right] \qquad (2)$$

where R_k^{-1} exists, $|R_k|$ is the determinant of the matrix R_k and the superscript T denotes the transpose of a vector or a matrix. For simplicity ,let us assume that

$$R_1 = R_2 = R_3 = \ldots\ldots\ldots\ldots = R_K = R$$

and

$$P_1 = P_2 = P_3 = \ldots\ldots\ldots\ldots = P_K = P$$

From equations (1) and (2), an unknown pattern vector $\underline{X}$ is classified into C_j if

$$f_j(\underline{X}) \geq f_k(\underline{X}) \qquad \text{for all } k \neq j$$

or in other words

$$\exp.\left[(\underline{X}-\underline{m}_j)^T R^{-1} (\underline{X}-\underline{m}_j)\right] \geq \exp.\left[(\underline{X}-\underline{m}_k)^T R^{-1} (\underline{X}-\underline{m}_k)\right] \qquad (3)$$

$$\text{for all } k \neq j$$

Since the logarithm is a monotonically increasing function, then equation (3) may be expressed as

$$(\underline{X}-\underline{m}_j)^T R^{-1} (\underline{X}-\underline{m}_j) \geq (\underline{X}-\underline{m}_k)^T R^{-1} (\underline{X}-\underline{m}_k) \qquad \text{for all } k \neq j \qquad (4)$$

Now, define a discriminant function $D_j(\underline{X})$ as

$$D_j(\underline{X}) = (\underline{X} - \underline{m}_j)^T R^{-1} (\underline{X} - \underline{m}_j) \quad (5)$$

Equation (5) may be expressed in linear form (neglecting 2nd order term) as:

$$D_j(\underline{X}) = \underline{h}_j^T \underline{X} + a_j \quad (6)$$

where

$$\underline{h}_j^T = -2\, \underline{m}_j^T R^{-1} \qquad \text{and} \qquad a_j = \underline{m}_j^T R^{-1} \underline{m}_j$$

Using Eq. (6), an unknown pattern vector $\underline{X}$ is classified into C_j if

$$D_j(\underline{X}) \geq D_k(\underline{X}) \qquad \text{for all } k \neq j \quad (7)$$

The decision surface between C_j and C_k is defined by

$$D_j(\underline{X}) - D_k(\underline{X}) = 0 \quad (8)$$

A block diagram for Bayes classifier is shown in Fig. 1.

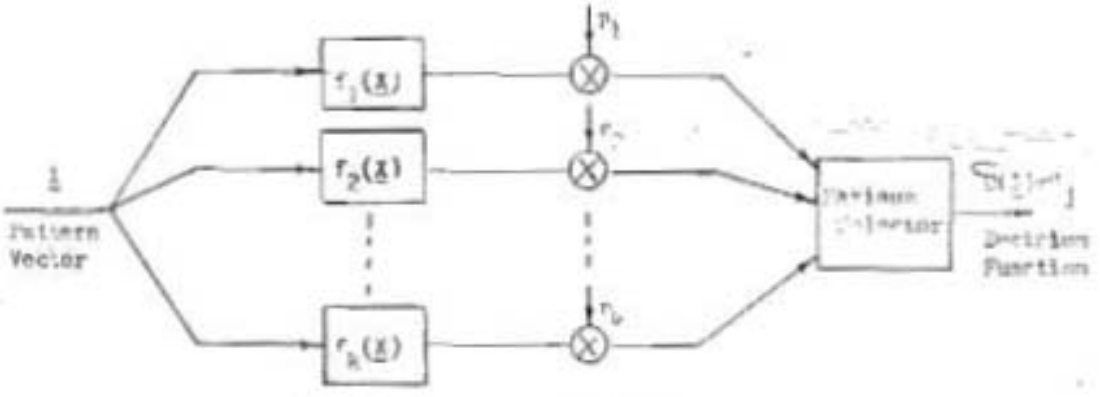

Fig.1, Bayes Classifier

III-PERCEPTRON CLASSIFIER

On this type of classifiers, no assumption on probability distributions of pattern vectors are made. The only knowledge available is a sequence of training samples $\underline{X}_1$, $\underline{X}_2$, ... $\underline{X}_n$ with the classification of each sample being known. The sample size n is finite, and the dimensionality N of the pattern space Ω is much less than n.

The method discussed here does not estimate the parameters of the probability density functions, hence it is considered as a nonparametric scheme. When the probability density functions are known, the general form of the discriminant functions may be subjectively decided. The choice of this general form

may be influenced by several factors: the allowable complexity of the machine, the desired classification accuracy, and dimensionality and sample size. Once the general form is decided, the specific discriminant functions for a particular pattern recognition problem are determined by the machine through its learning phase using the training samples.

The simplest form of discriminant function is linear. Linear discriminant functions are defined as

$$D_k(\underline{X}) = \alpha_{1k} x_1 + \alpha_{2k} x_2 + \ldots\ldots\ldots \alpha_{Nk} x_N + \alpha_{N+1,k} \quad ; k=1, 2, \ldots K \qquad (9)$$

where K is the number of classes, $x_1, x_2, \ldots\ldots x_N$ are the N components of the pattern vector $\underline{X}$, and $\alpha_{1k}, \alpha_{2k}, \ldots\ldots \alpha_{N+1,k}$ are weighting coefficients. Using vector notations, Eq.(9) may be expressed as

$$D_k(\underline{X}) = \hat{\underline{X}}^T \underline{\alpha}_k = \underline{\alpha}_k^T \hat{\underline{X}} \qquad (10)$$

where

$$\hat{\underline{X}}^T = [\underline{X}^T, 1]$$

is the transpose of $\hat{\underline{X}}$ called augmented pattern vector, and $\underline{\alpha}_k$ is the kth weighting vector consisting of the N+1 weighting coefficients.

A machine that employs linear discriminant functions is called a linear machine or perceptron [2]. The K weighting vectors $\{\underline{\alpha}_i , i=1,2,\ldots\ldots K\}$ for the different classes are usually estimated by the machine using the training samples. Given a training set $S_x = \{\underline{X}_1 , \underline{X}_2 , \ldots\ldots \underline{X}_m\}$, a training sequence S_{xx} is constructed on the set S_x, such that each $\underline{X}$ in S_{xx} is a member of S_x and every member of S_x occurs with infinite frequency in S_{xx}. Having done that, a training sequence $S_{\hat{x}\hat{x}}$ is formed by replacing each $\underline{X}$ in S_{xx} by its augmented vector $\hat{\underline{X}}$. Let $\hat{\underline{X}}_i$ be the ith member of the training sequence $S_{\hat{x}\hat{x}}$, and $\underline{\alpha}_k(i)$ be the estimate of the kth weighting vector at the ith iteration, where k=1, 2, 3,.....K. Suppose $\underline{X}_i$ belongs to C_j, then the training algorithm is stated as: if $\hat{\underline{X}}_i$ is correctly classified to C_j, i.e, if

$$\underline{\alpha}_j^T(i)\, \hat{\underline{X}}_i \geqslant \underline{\alpha}_k^T(i)\, \hat{\underline{X}}_i \qquad \text{for all } k \neq j \qquad (11)$$

then

$$\underline{\alpha}_k(i+1) = \underline{\alpha}_k(i) \qquad , k = 1, 2, 3, \ldots\ldots\ldots K \qquad (12)$$

i.e, no correction is performed. On the other hand, if

$$\underline{\alpha}_j^T(i)\, \hat{\underline{X}}_i \quad \underline{\alpha}_\ell^T(i)\, \hat{\underline{X}}_i \qquad (13)$$

where

$$\underline{\alpha}_\ell^T(i)\, \hat{\underline{X}}_i = \max.[\underline{\alpha}_1^T(i)\hat{\underline{X}}_i , \underline{\alpha}_2^T(i)\hat{\underline{X}}_i , \ldots\ldots \underline{\alpha}_k^T(i)\hat{\underline{X}}_i] \text{ for all } \ell \neq j \qquad (14)$$

then $\underline{w}_j(i)$ and $\underline{w}_\ell(i)$ need only be adjusted, i.e.,

$$\begin{aligned} \underline{w}_j(i+1) &= \underline{w}_j(i) + \hat{\underline{x}}_i \\ \underline{w}_\ell(i+1) &= \underline{w}_\ell(i) - \hat{\underline{x}}_i \qquad \text{for all } k \neq j \text{ and } k \neq \ell \\ \underline{w}_k(i+1) &= \underline{w}_k(i) \end{aligned} \tag{15}$$

The inequality in Eq.(15) says that $\hat{\underline{x}}_i$ is incorrectly classified into C_ℓ, and the equals sign is included because of the definition of linear separability [2].

Once the training or learning has been performed, the machine classifies the unknown pattern vectors using the decision rule:

$$\underline{x} \in C_j \quad \text{if:}$$

$$D_j(\underline{x}) > D_k(\underline{x}) \qquad \text{for all } k \neq j \tag{16}$$

where $D(\underline{x})$ is the decision function. The decision surfaces are specified by

$$D_j(\underline{x}) - D_k(\underline{x}) = 0 \tag{17}$$

A block diagram of the perceptron is shown in Fig.2.

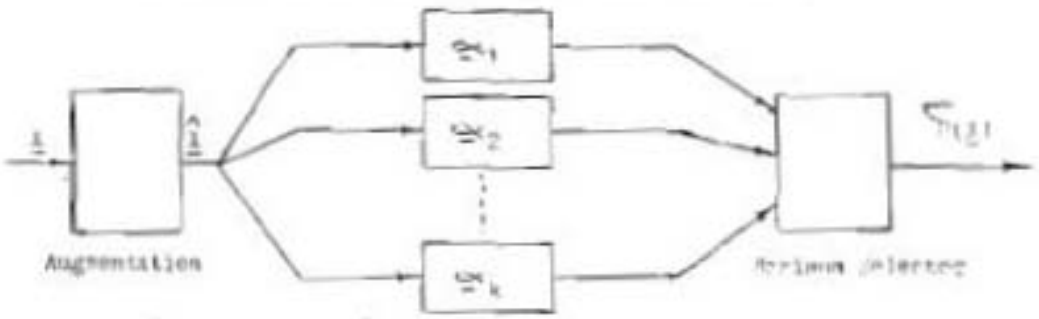

Fig. 2, Perceptron Classifier

IV- UNSUPERVISED AVERAGE CLASSIFIER:

Again, no assumptions are made on the probability distribution or density function of the pattern vectors. The only provided knowledge is a sequence of K- sets of training pattern vectors with known classification. The number of vectors in each class (each set) must be greater than the dimensionality of the pattern space N.

In order to clarify the idea behined this technique, consider the case shown in Fig.3. Two classes C_1 and C_2 in a two dimensional pattern space are shown. The ensemble average pattern vectors $\underline{m}_1$ and $\underline{m}_2$ are estimated for C_1 and C_2 respectively.

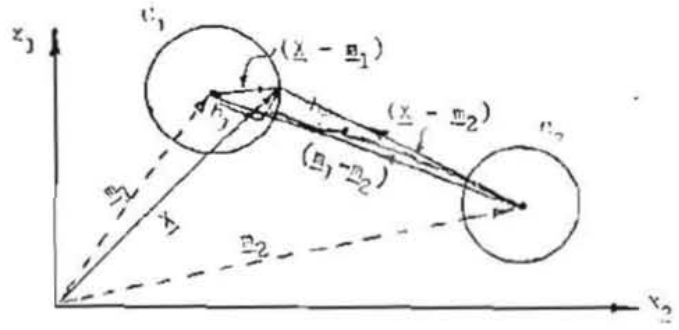

Fig. 3, Ensemble Average Classifier
(An Example)

In order to classify an unknown pattern vector $\underline{X}$ into either C_1 or C_2, project the vectors $(\underline{X} - \underline{m}_1)$ and $(\underline{X} - \underline{m}_2)$ on the ensemble average difference vector $(\underline{m}_2 - \underline{m}_1)$. Let these projections be h_1 and h_2, then $\underline{X}$ belongs to C_1 if

$$h_1^2 \geqslant h_2^2 \tag{18}$$

otherwise $\underline{X}$ belongs to C_2 .

To generaline the above procedure, let $C = \{C_i \ , \ i = 1, 2, \ldots\ldots K\}$ be a set of K classes, in each class there is a set of training pattern vectors belonging to it. Classifying an unknown pattern vector $\underline{X}$ into one of the K classes is carried out as follows:

i-Estimate the ensemble average pattern vector for each class as:

$$\underline{m}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \underline{X}_j \quad ; \qquad i = 1, 2, \ldots\ldots K \tag{19}$$

where n_i is the number of training pattern vectors in C_i and $\underline{X}_j$, $j = 1, 2, \ldots\ldots n_i$ are the pattern vectors in C_i.

ii-An unknown pattern vector $\underline{X}$ is classified into C_i if

$$h_i^2 \geqslant h_j^2 \qquad \text{for all } j \neq i \tag{20}$$

or

$$\left[(\underline{X} - \underline{m}_i)^T(\underline{m}_i - \underline{m}_j)\right]^2 \geqslant \left[(\underline{X} - \underline{m}_j)^T(\underline{m}_i - \underline{m}_j)\right]^2 \tag{21}$$

After some algebriac manipulations, Eq.(21) may be expressed as:

$$\underline{X}^T(\underline{m}_i - \underline{m}_j) \leqslant T_{ij} \tag{22}$$

where T_{ij} is a threshold value defined as

$$T_{ij} = \frac{(\underline{m}_i^T \underline{m}_i)\left[\underline{m}_i^T(\underline{m}_i - 2\underline{m}_j)\right] - (\underline{m}_j^T \underline{m}_j)\left[\underline{m}_j^T(\underline{m}_j - 2\underline{m}_i)\right]}{2(\underline{m}_i^T \underline{m}_i + \underline{m}_j^T \underline{m}_j - 2\underline{m}_i^T \underline{m}_j)} \tag{23}$$

for all i , j = 1, 2, ... K

Note that $T_{ij} = -T_{ji}$, a property that provide a great reduction in threshold

value calculations.

iii-With the aid of Eq.(23) and inequality (22), an unknown pattern vector $\underline{X}$ is classified into C_1 or C_2

iv-Here, depending on the result of step iii, $\underline{X}$ is classified into either C_3 or the class chosen in step iii.

v-Repeat step iv for C_4 , C_5 ..... C_K.

After (K-1) steps, the machine reaches a decision and assigns the unknown vector $\underline{X}$ into the correct class.

IV- PERFORMANCE COMPARISON

In this section the three classifier discussed earlier are compared from the points of view of learning cost, computational complexity, and memory size required for data storage.

1-Machine Learning

It has been stated in section II that Bayes classifier is designed under some restrictions. These are:

a)the probability distribution and probability density function must be known aperiori for all classes.

b)equal misclassification cost is assumed for all classes.

c)probability density function is restricted to Gaussian distribution with identical covariance matrix for all classes.

From the above discussion it can easily be seen that the Bayes classifier, although optimal, is designed under great restrictions and requires matrix inversion.

Perceptron classifier requires a large number of computations through a recursive algorithm to prepare the library of the machine. This number of computations depends on the initial guess $\underline{C}(0)$ and verification of the assumption of linear seperability.

Ensemble average classifier is designed under no restrictions, such as linear seperability or aperiori probability. Machine learning requires a limited number of computations and no matrix inverrion is involved.

2-Computation Cost

The account of computations required to classify an unknown pattern give an indication of the speed and cost of classifier used. Table 1, shows the number of multiplications and additions performed by the three classifiers discussed in this paper.

Table 1. Number of Multiplications and Additions Required for Classification of Unknown Pattern.

Classifier	Number of Additions	Number of Multiplications
Bayes	KN	KN
Perceptron	KN	KN
Ensemble Average	(2N-1)(K-1)	(K-1)N

It can be seen from table 1, that ensemble average classifier requires greater number of additions than both Bayes and perceptron classifiers. On the

other hand it requires less number of multiplications than the others. On average, one may say that the three classifiers require almost the same number of computations.

3-Memory Size

Table 2, Number of Memory Words Required for Data Storage.

Classifier	Number of Data Words
Bayes	$K(N+1)$
Perceptron	$K(N+1)$
Ensemble Average	$KN + \frac{1}{2}K(K-1)$

Table 2, shows the memory size required for library storage in the three classifiers. It is evident from this table that the ensemble average classifier requires greater memory size than both Bayes and perceptron classifiers. This increased memory size costs nothing if compared to the complexity of Bayes and perceptron classifiers.

4- EXPERIMENTAL RESULTS

A series of computer simulation experiments has been carried out to compare the performance of the three classifiers under study. For a matter of convenience, only three experiments are reported here. These experiments are:

Experiment No.1

In this experiment, two classes of Gaussian distribution are considered as training data. Class C_1 has

$$\underline{m}_1 = [\,0 \quad 1\,]^T \,, \quad R_1 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

and class C_2 has

$$\underline{m}_2 = [\,1 \quad 0\,]^T \,, \quad R_2 = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix}$$

These classes are shown in Fig. 4.

Since the covariance matrices R_1 and R_2 are not equal, then linear Bayesian classifier is not feasible. Moreover, the two classes are not linearly seperable, therfore, conversion never occured while designing the perceptron classifier coefficients $\{\alpha_i, i=1,2,\ldots\ldots K\}$.

For the current experiment, it was easy to design the ensemble average classifier. Table 3, shows number of patterns from each class used for machine learning and the corresponding percentage recognition rate when the same patterns

used as test patterns.

Table 3, Results Obtained by Ensemble Average Classifier (for Experiment No.1)

No. of Training Patterns Used for Machine Learning	Percentage Recognition Rate
10	66.5
50	61.5
100	67.0
200	65.5
500	66.0
Average Recognition Rate = 64.5 %	

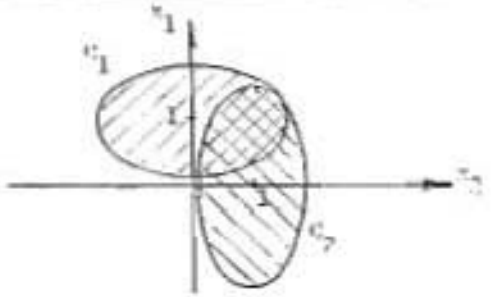

Fig.4. Data for Experiment No.1

Experiment No. 2

Here, two classes with Gaussian distribution are chosen with mean vectors and covariance matrices as:

$$\underline{m}_1 = [2 \quad 6]^T \quad , \quad \underline{m}_2 = [6 \quad 2]^T$$

and

$$R_1 = R_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

For this experiment, it was easy to design the three classifiers. They provided equal recognition rate of 100 % when training data were used as test data and 99% when completely strange data were used. This experiment was repeated for three dimensional instead of two dimensional pattern space. The mean vectors and the covariance matrices were chosen as:

$$\underline{m}_1 = [-3 \quad 5 \quad 1]^T \quad , \quad \underline{m}_2 = [3 \quad -3 \quad 1]^T$$

and

$$R_1 = R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Results similar to those obtained in the previous experiment were recorded.

Experiment No. 3

This experiment was designed to compare between efficiency of Bayesian and ensemble average classifiers when the classes are not linearly separable. Two classes of Gaussian distribution are chosen. The covariance matrices are given as:

$$R_1 = R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and class C_1 has a fixed mean vector of $\underline{m}_1 = [0 \quad 0 \quad 0]^T$ while C_2 has a varying mean vector, i.e, $\underline{m}_2 = [1 \quad 0 \quad 0]^T$, $\underline{m}_2 = [2 \quad 0 \quad 0]^T$ .....etc

For a given distance between the two classes, both Bayesian and ensemble average classifiers provided identical recognition rates. Table 4, shows the percentage recognition rate as a function of the distance between the two classes.

Table 4, Results of Experiment 3

Distance Between the Two Classes	Percentage Recognition Rate
1	68
2	81
3	92
4	96.5
5	99.5
6	99.5
7	100
8	100

VI- CONCLUSION AND PRACTICAL APPLICATIONS

In this paper, the ensemble average classifier was advised and tested. Although, it requires a bit higher memory size, it is much simpler, faster, and easier to design. From the point of view of recognition rate, it has been found by experiments that the new classifier stands side by side with the Bayesian and

perceptron classifiers. Moreover, in cases where Bayesian and perceptron were not feasible, the ensemble average classifier provided a recognition rate as high as 66%.

The variety of possible applications of automatic classifications is: ranging over automatic control, medical diagnoses, psychological learning, pattern recognition, and image analysis. We are currently applying the ensemble average method to pattern recognition, specifically, the hanwritten arabic characters [6].

REFERENCES

1- Duda,R.O. and Hart,P.E., "Pattern Classification and Scene Analysis",Wiley, New York, 1973.
2-Young,T.Y., and Calvert,T.W., "Classification, Estimation and Pattern Recognition," American Elsevier Publishing Co., New York, London, 1974.
3- Anderson,T.W., "An Introduction to Multivariate Statistical Analysis," Wiley, New York, 1958.
4- Cover, T.M., and Hart,P.E.,"Nearest Neighbor Pattern Classification," IEEE Trans. Inform. Theory, Vol. IT-13, PP21 - 27, Jan. 1967.
5- Rosenblatt, F.,"The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Rev., Vol. 65, No.6,PP386-408, 1958.
6- Zaki,F.W., El-Kennaly,S.N.,Abd El-Fattah,A.I.,and Knab,Y.M., "A New Technique for Arabic Handwritten Recognition," Proc. of the 11th Int. Conf. for Statis., Comp. Science, Social and Demographic Research, 29 March- 3 April,1986, Ain Shams Univ., Cairo, Egypt, PP 171 - 180.