Classification Algorithms in Pattern Recognition

GEORGE NAGY, MEMBER, IEEE

Abstract—Linear and nonlinear methods of pattern classification which have been found useful in laboratory investigations of various recognition tasks are reviewed. The discussion includes correlation methods, maximum likelihood formulations with independence or normality assumptions, the minimax Anderson–Bahadur formula, trainable systems, discriminant analysis, optimal quadratic boundaries, tree and chain expansions of binary probability density functions, and sequential decision schemes. The area of applicability, basic assumptions, manner of derivation, and relative computational complexity of each algorithm are described. Each method is illustrated by means of the same two-class two-dimensional numerical example. The "training set" in this example comprises four samples from either class; the "test set" is the set of all points in the normal distributions characterized by the sample means and sample covariance matrices of the training set. Procedural difficulties stemming from an insufficient number of samples, various violations of the underlying statistical models, linear nonseparability, noninvertible covariance matrices, multimodal distributions, and other experimental facts of life are touched on.

Manuscript received August 20, 1967. This paper was presented at the 1967 Conference on Speech Communication and Processing, Cambridge, Mass.

The author is with the IBM Watson Research Center, Yorktown Heights, N. Y. He is currently on leave at the University of Montreal, Montreal, Canada.

Introduction

ALTHOUGH research in pattern recognition has not yet solved many of the problems which were thought to be within easy reach ten years ago, there has been sufficient progress to encourage abandonment of ad hoc design techniques in some practical applications. This survey is intended as a guide to some of the more commonly used classification algorithms. Only methods where the parameters are automatically derived from identified samples are considered.

In the hope that some uniformity of viewpoint will be appreciated by workers only peripherally involved in pattern recognition, the discussion will emphasize geometric concepts wherever possible. Alternative vocabulary and notation may be drawn from statistical decision theory, the logical algebra of switching circuits, linear programming, set theory, communication theory, and nerve net studies. Which of these disciplines will contribute most to the eventual emergence of a cohesive theory of pattern recognition remains to be seen.

The primary goal in designing a pattern classifier is to have it perform well (achieve a high recognition rate) on new data. When the training data are truly representative of the test data, and when a very large number of training patterns are available, it is usually argued that it is sufficient to design the classifier to perform adequately on the training set. In practice, the training set is always too small, and this argument is, therefore, fallacious.

With some a priori information about the nature of the underlying probability distributions, it is, indeed, possible to predict from a limited training set the performance on the test set. In real problems, however, even the probability model must be inferred from the training set. In the face of this dilemma, the reader must be cautioned that it is possible to overdesign the classifier by tailoring it too closely to the training set at the expense of performance on the test set. Matching the design method to the number of samples available is not easy, but a simple rule of thumb is that the more complicated the method, the more samples are required.

In the following discussion, the ith pattern will be treated as a column vector $\bar{x}_i$; $\bar{x}_i'$ is the corresponding row vector. The components x_{ij} of $\bar{x}_i$ denote individual observations: the energy around 300 Hz in the first 100 milliseconds of an utterance, whether the lower left-hand corner of a character is black or white, the temperature of a hospital patient, the location of the peak in an electrocardiogram. In some problems, the choice of observations is critical. In others, a natural set of coordinates, such as the gray levels in the matrix representation of a photograph, exists.

One important distinction between pattern recognition and other related disciplines, such as automatic control theory, switching theory, and statistical hypothesis testing, is the high dimensionality of the vectors $\bar{x}_i$. Were it not for the fact that $\bar{x}_i$ typically runs to

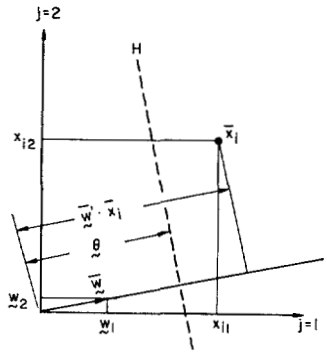

Fig. 1. A linear categorizer. The normalized weight vector $\bar{w}$ and the normalized threshold Θ define a two-class linear categorizer. The pattern $\bar{x}_i$ is assigned to class C^1 because its projection $\bar{w}'\bar{x}_i$ on the normalized weight vector is greater than Θ. Any pattern on the same side of hyperplane H as $\bar{x}_i$ would be assigned to C^1.

hundreds of components, with hundreds, thousands, or even millions (as in the bank font problem) of samples, we could undertake calculations far more sophisticated than those discussed in this paper.

In the beginning, we will confine our attention to two-class problems. In principle, any multiclass problem can be treated as a number of two-class problems involving the separation of each class from the remainder of the universe, but this does not, as a rule, lead to the most economical solution. The class will always be denoted by a superscript.

All of the categorization methods where the components of the pattern vector are not limited to binary numbers will be illustrated by means of the same two-dimensional example. In this example, there are eight patterns in the training set, four from each class.

Class C^1

$$\bar{x}_1 \quad \bar{x}_2 \quad \bar{x}_3 \quad \bar{x}_4$$
$$\begin{pmatrix}0\\2+2\sqrt{2}\end{pmatrix}\begin{pmatrix}0\\2-2\sqrt{2}\end{pmatrix}\begin{pmatrix}\sqrt{6}\\2\end{pmatrix}\begin{pmatrix}-\sqrt{6}\\2\end{pmatrix}$$

Class C^2

$$\bar{x}_5 \quad \bar{x}_6 \quad \bar{x}_7 \quad \bar{x}_8$$
$$\begin{pmatrix}4\\2\sqrt{2}\end{pmatrix}\begin{pmatrix}4\\-2\sqrt{2}\end{pmatrix}\begin{pmatrix}4+\sqrt{2}\\0\end{pmatrix}\begin{pmatrix}4-\sqrt{2}\\0\end{pmatrix}.$$

In order to compare the performance of the various methods, it is assumed that these samples originate from multivariate normal (or Gaussian) populations with means $\bar{\mu}^k$ equal to the sample means, and covariance matrices A^k equal to the sample covariance matrices:

$$\bar{\mu}^1 = \begin{pmatrix}0\\2\end{pmatrix}, \qquad \bar{\mu}^2 = \begin{pmatrix}4\\0\end{pmatrix},$$
$$A^1 = \begin{pmatrix}3 & 0\\0 & 4\end{pmatrix}, \qquad A^2 = \begin{pmatrix}1 & 0\\0 & 4\end{pmatrix}.$$

The samples, and the elliptical equiprobability contours of the normal density functions

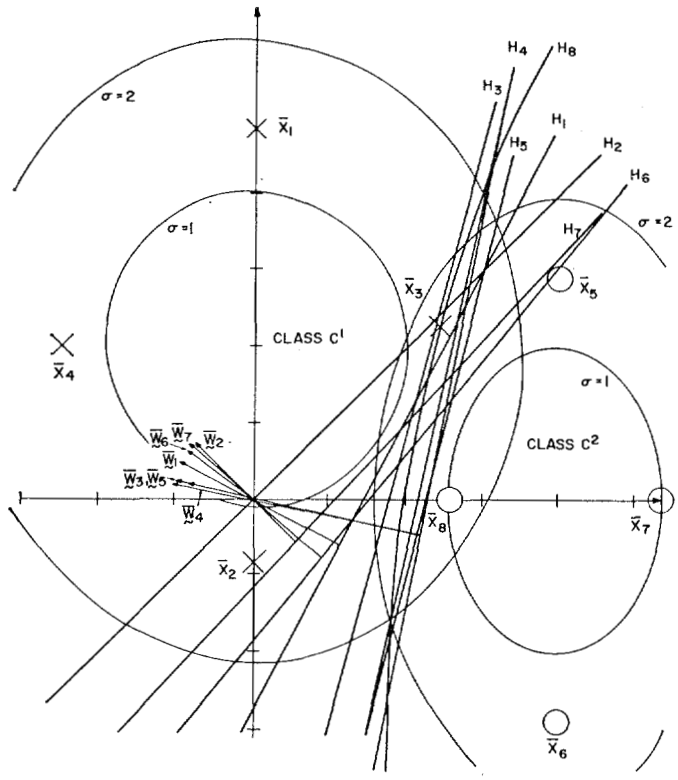

Fig. 2. Common types of linear categorizers. The x's and the ○'s indicate the training samples in Classes C^1 and C^2, respectively. The ellipses are the equiprobability contours on the postulated distributions in the test data. The subscripts associated with the hyperplanes and weight vectors pertain to the following categorizers: 1) distance to means; 2) correlation; 3) approximate maximum likelihood; 4) Anderson–Bahadur; 5) discriminant analysis; 6) approximate discriminant analysis; 7) trainable machine; 8) optimal quadratic boundary.

$$p^k(\bar{x}) = (2\pi)^{-n/2}(\det A^k)^{-1/2} \cdot \exp\left[-\tfrac{1}{2}(\bar{x}-\bar{\mu}^k)'A^{k-1}(\bar{x}-\bar{\mu}^k)\right]$$

with the above parameters, are shown in Fig. 2.

Linear Classification

A linear categorizer assigns an unknown pattern $\bar{x}_i$ to class C^1 if $\bar{x}_i'\cdot\bar{w}\geq\theta$, and to class C^2 otherwise. The coefficients w_j of $\bar{w}$ are proportional to the components of a vector (through the origin) onto which the patterns are projected. In the two-dimensional example in Fig. 1, all the points which are to the right of the dotted straight line perpendicular to the vector

$$\bar{w}\left(\frac{w_1}{\sqrt{w_1^2+w_2^2}}, \frac{w_2}{\sqrt{w_1^2+w_2^2}}\right)$$

at a distance $\theta/\sqrt{w_1^2+w_2^2}$ from the origin are assigned to class C^1.

When $\bar{x}_i$ has more than two components, it is still projected onto the vector $\bar{w}$, but now a hyperplane, rather than a line, separates the classes. The vector $\bar{w}$ is traditionally referred to as the weight vector, because its components represent the relative importance of each observation in deciding the class assignment.

TABLE I

COMPARISON OF DECISION METHODS

No. in Fig. 2	Method	Parameters				% Error Rate on Test Data		
		Unnormalized		Normalized				
		$\bar{w}$	θ	$\bar{w}$	θ	On C^1	On C^2	Average
1	Distance from means	−4.00 2.00	−6.00	−0.89 0.45	−1.34	11.5	3.8	7.7
2	Dot product with normalized means	−1.00 1.00	0.00	−0.71 0.71	0.00	22.7	3.7	13.2
3	Approximate maximum likelihood	−2.00 0.50	−3.50	−0.96 0.24	−1.70	10.7	2.2	6.5
4	Anderson–Bahadur	−2.27 .50	−5.20	−0.98 0.21	−2.26	5.9	5.9	5.9
5	Discriminant analysis	−4.00 1.00	−9.15	−0.97 0.24	−2.21	6.3	6.3	6.3
6	Approximate discriminant analysis	−1.28 1.00	−3.10	−0.79 0.61	−1.21	4.4	19.8	12.1
7	Trainable machine	−3.89 3.85	−4.00	−0.72 0.70	−0.73	12.5	9.2	10.8
8	Optimum quadratic boundary							~5.6

The weights w_j and the threshold θ may be multiplied by any constant without a change in the resulting classification. It is customary to normalize the weight vector to unit length, as shown.

In the two-dimensional example we propose to discuss, the test set is assumed to consist of an infinite number of patterns with known Gaussian distributions. The error for any linear categorizer would be computed by projecting these distributions onto the weight vector. The projected one-dimensional distributions are also Gaussian, with means $\bar{w}'\bar{\mu}^k$ and variances $\bar{w}'A^k\bar{w}$. The error rate can be readily found from a table of cumulative Gaussian probabilities:

$$P(\text{error}) = \frac{1}{2}\phi\left[\frac{\theta - \bar{w}'\bar{\mu}^1}{(\bar{w}'A^1\bar{w})^{1/2}}\right] + \frac{1}{2} - \frac{1}{2}\phi\left[\frac{\theta - \bar{w}'\bar{\mu}^2}{(\bar{w}'A^2\bar{w})^{1/2}}\right]$$

where

$$\phi(y) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y} e^{-y^2/2}dy.$$

The weight vectors, hyperplanes, and other decision boundaries obtained by the various categorization procedures about to be described are shown in Fig. 2. The computed error rates, which appear in Table I, should be considered only as numerical illustrations of the formulas. In a real problem, additional errors would result from imperfect estimates of the population parameters.

Piecewise linear categorizers simply contain more than one weight vector to characterize a class pair. Boundaries of arbitrary complexity may be approximated by the resultant profusion of hyperplanes.[11], [35]

CORRELATION

The simplest method to compute the parameters of the linear categorizer is to let $\bar{w} = \bar{c}^1 - \bar{c}^2$, where $\bar{c}^1$ and $\bar{c}^2$ represent "typical" members of the two classes. Customarily, $\bar{c}^1$ is set equal to $\bar{\mu}^1$, the centroid of class C^1, and $\bar{c}^2 = \bar{\mu}^2$, the centroid of C^2. Thus,

$$\left.\begin{aligned} &\bar{w} = \bar{\mu}^1 - \bar{\mu}^2 \\ \text{and} \\ &\theta = \tfrac{1}{2}(\bar{\mu}^1 - \bar{\mu}^2)'(\bar{\mu}^1 + \bar{\mu}^2). \end{aligned}\right\} \quad (1)$$

The resulting hyperplane is H_1 in Fig. 2. When it is felt that the magnitude of the feature vectors matters less than their orientation, $\bar{w}$ is set equal to

$$\frac{\bar{\mu}^1}{|\bar{\mu}^1|} - \frac{\bar{\mu}^2}{|\bar{\mu}^2|},$$

with a threshold of 0. This is hyperplane H_2 in Fig. 2. Such a decision procedure would be useful, for example, in classifying sustained sounds, where the overall intensity depends only on the distance from the microphone, and the sound is fully characterized by the relative intensities of the various frequencies.

The correlation method is often used on binary data, where it is sometimes referred to as mask or template matching. For computational efficiency, the number of mismatching, rather than matching, bit positions is usually computed, and the calculation is truncated when this reaches a preset threshold. The use of additive

and multiplicative constants in normalization sometimes results in startling changes in performance.

Theoretically, the correlation methods can be shown to be optimal only under certain very restrictive symmetry conditions on the distributions.

Maximum Likelihood

The application of the maximum likelihood principle to pattern classification makes use of Bayes' formula for conditional probabilities to show that in order to determine the greater of $P[C^1|\bar{x}_i]$ and $P[C^2|\bar{x}_i]$, it is sufficient to compare $P[\bar{x}_i|C^1]$ and $P[\bar{x}_i|C^2]$ (for equal a priori probabilities on the classes). The determination of the conditional probability of the pattern vector, given the class, leads to a linear expression in the components of the pattern vector under several assumptions.

When the components x_{ij} are statistically independent of one another,

$$P[\bar{x}_i|C^k] = \prod_j P[x_{ij}|C^k].$$

Geometrically, this condition corresponds to having the principal axes of the two distributions parallel to the coordinate axes. If, in addition, the x_{ij}'s are binary, it can be shown[29] that

$$P[C^1|\bar{x}_i] \geq P[C^2|\bar{x}_i] \quad \text{if and only if} \quad \bar{x}_i' \cdot \bar{w} \geq \theta,$$

where

$$\left.\begin{aligned} w_j &= \ln \frac{P[x_{ij}=1|C^1]P[x_{ij}=0|C^2]}{P[x_{ij}=1|C^2]P[x_{ij}=0|C^1]} \\ \text{and} \\ \theta &= \sum_j \ln \frac{P[x_{ij}=0|C^2]}{P[x_{ij}=0|C^1]}. \end{aligned}\right\} \quad (2)$$

Despite the fact that the independence assumption is very seldom satisfied in practice, this decision method has been widely used in cases where the high dimensionality of the pattern vectors precludes more complicated calculations. Let us compute the parameters and the error rate using this scheme in another two-dimensional example (where the patterns are restricted to binary components).

Class	Pattern	Number of Patterns in Training Sample
C^1	(0, 0)	60
	(1, 1)	40
C^2	(1, 0)	30
	(0, 1)	70

The probabilities needed to estimate the components of the weight vector and the threshold are readily calculated:

$$P[x_{i1}=1|C^1] = 0.4, \qquad P[x_{i2}=1|C^1] = 0.4,$$
$$P[x_{i1}=1|C^2] = 0.3, \qquad P[x_{i2}=1|C^2] = 0.7.$$

The weight vector and the resulting hyperplane calculated from (2) are shown on Fig. 3. Since the independence assumption is clearly violated, we need not be surprised that the 0.35 error rate obtained with this plane on the training sample is higher than that obtained with other planes. The plane shown in dotted lines on Fig. 3, for example, yields only 15 percent errors. With a nonlinear scheme, we could separate the classes without error, since there is no overlap between the distributions.

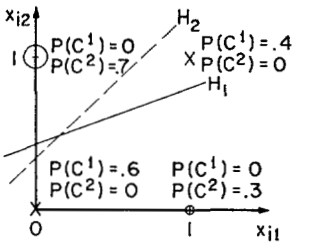

Fig. 3. Categorizers for binary patterns. Class 1 is indicated by circles, Class 2 by crosses. The size of the symbols is proportional to the postulated probability density distribution. H_1 is the hyperplane calculated from (2); it is not as good as H_2, which could be obtained by inspection because of the low dimensionality of the problem. The unnormalized weight vector corresponding to H_1 is (0.441, −1.25), with a threshold of −0.41.

With a finite number of samples in the training set, it may happen that a numerator or a denominator in (2) vanishes. To avoid this problem, there is some theoretical justification for estimating $P[x_{ij}=1|C^k]$ by means of $(M_j^k+1)/(N^k+2)$, instead of M_j^k/N^k as above, where M_{jk} is the number of samples in class k with jth component equal to 1, and N^k is the total number of samples in class k.[15]

Another instance where the maximum likelihood classifier is linear is in the case of normal distributions with identical covariance matrices. The form of the distributions is again

$$p^k(\bar{x}) = \frac{1}{(2\pi)^{n/2}|A|^{1/2}} \exp\left[-\tfrac{1}{2}(\bar{x}-\bar{\mu}^k)'A^{-1}(\bar{x}-\bar{\mu}^k)\right],$$

where the elements of the matrix A are the same whichever class k is used to derive them.

$$A = \underset{k}{E}[(\bar{x}-\bar{\mu}^k)(\bar{x}-\bar{\mu}^k)']$$

and

$$\bar{\mu}^k = \underset{k}{E}[\bar{x}].$$

The assumption of equal covariances is reasonable, for example, in digitized photographs, where the adjacent cells are likely to be positively correlated regardless of class, because the gray scale seldom contains rapid transitions.

In comparing the ratio of the probability distribution functions to 1, the exponents subtract and the second-order terms in the pattern components cancel. In the resulting linear expression,

$$\left.\begin{aligned} \bar{w} &= (\bar{\mu}^1-\bar{\mu}^2)'A^{-1} \\ \text{and} \\ \theta &= \tfrac{1}{2}(\bar{\mu}^1-\bar{\mu}^2)'A^{-1}(\bar{\mu}^1+\bar{\mu}^2). \end{aligned}\right\} \quad (3)$$

To apply this method to our example, we shall approximate A by the mean of the covariance matrices for the two classes. Thus,

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}, \quad \text{and} \quad A^{-1} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{4} \end{pmatrix}.$$

The resulting hyperplane is H_3 in Fig. 2.

If the distributions are spherical, the covariance matrix is proportional to the identity matrix, and the weight vector corresponds to the difference of the centroids, as in simple correlation.

When the number of samples in the training set is small compared to the dimensionality of the pattern vectors, the covariance matrix may be singular. Without the inverse, it is necessary either to guess the values of the variance in the missing directions, or to confine the solution weight vector to the subspace actually spanned by the training set.[31],[33]

Minimax Decision Rule—The Anderson–Bahadur Formula

With normal distribution functions characterized by unequal covariance matrices, the maximum likelihood boundary is nonlinear. Instead, the minimax criterion, which equalizes the probabilities of the two kinds of errors (for equal a priori probabilities), is used.

The following implicit equation for the weight vector $\bar{w}$ has been derived by Anderson and Bahadur:[1]

$$\bar{w} = \left[\frac{(\bar{w}'A^2\bar{w})^{1/2}}{(\bar{w}'A^1\bar{w})^{1/2} + (\bar{w}'A^2\bar{w})^{1/2}} A^1 + \frac{(\bar{w}'A^1\bar{w})^{1/2}}{(\bar{w}'A^1\bar{w})^{1/2} + (\bar{w}'A^2\bar{w})^{1/2}} A^2 \right]^{-1} \cdot (\bar{\mu}^1 - \bar{\mu}^2) \quad (4)$$

where

μ^k = mean of class k, and
A^k = covariance matrix of class k.

Equation (4) can be solved with conventional iterative methods, using matrix inversion. When the dimensionality is high, it is desirable to avoid matrix inversion with the method of conjugate gradients, which is guaranteed to converge in, at most, a number of steps equal to the dimensionality.[18]

The probability of classification errors with the optimum threshold can be computed in terms of the weight vector

$$P(\text{error}) = 1 - \phi\left[\frac{(\bar{w}'A^1\bar{w})^{1/2}(\bar{w}'A^2\bar{w})^{1/2}}{(\bar{w}'A^1\bar{w})^{1/2} + (\bar{w}'A^2\bar{w})^{1/2}} \right]$$

where

$$\phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-(1/2)v^2} dy.$$

The Anderson–Bahadur hyperplane is H_4 in Fig. 2.

Discriminant Analysis

When the form of the probability density functions governing the distribution of the pattern vectors is not known at all, the minimum-error hyperplane cannot be specified analytically. In this case, it seems intuitively desirable to find at least the direction in which the projections of the samples of each class fall as far as possible from those of the other class, but the internal scatter of each class is minimized. This is the object of discriminant analysis.[32],[43]

More formally, we wish to maximize

$$\sum_{\substack{\bar{x}_i \in C1 \\ \bar{x}_j \in C2}} (\bar{w}\cdot\bar{x}_i - \bar{w}\cdot\bar{x}_j)'(\bar{w}\cdot\bar{x}_i - \bar{w}\cdot\bar{x}_j),$$

subject to the constraint

$$\sum_{\substack{\bar{x}_i \in C1 \\ \bar{x}_j \in C1}} (\bar{w}\cdot\bar{x}_i - \bar{w}\cdot\bar{x}_j)'(\bar{w}\cdot\bar{x}_i - \bar{w}\cdot\bar{x}_j) + \sum_{\substack{\bar{x}_i \in C2 \\ \bar{x}_j \in C2}} (\bar{w}\cdot\bar{x}_i - \bar{w}\cdot\bar{x}_j)'(\bar{w}\cdot\bar{x}_i - \bar{w}\cdot\bar{x}_j) = \text{Constant}.$$

It can be shown, by using Lagrange multipliers, that the vector $\bar{w}$ which fulfills these conditions is the eigenvector associated with the largest eigenvalue λ of

$$(BA^{-1} - \lambda I)\bar{w} = 0 \quad (5)$$

where A is the intraclass sample scatter matrix,

$$A = \sum_{\bar{x}_i \in C1} (\bar{x}_i - \bar{\mu}^1)(\bar{x}_i - \bar{\mu}^1)' + \sum_{\bar{x}_i \in C2} (\bar{x}_i - \bar{\mu}^2)(\bar{x}_i - \bar{\mu}^2)'$$

and B is the interclass sample scatter matrix,

$$B = \sum_{\text{all } i} \left(\bar{x}_i - \frac{\bar{\mu}^1 + \bar{\mu}^2}{2} \right)\left(\bar{x}_i - \frac{\bar{\mu}^1 + \bar{\mu}^2}{2} \right)' - A.$$

The solution vector $\bar{w} = A^{-1}(\bar{\mu}^1 - \bar{\mu}^2)$ is identical to the approximation we used to the maximum likelihood solution in (3), but explicit use of the individual covariance matrices to equalize the two types of errors leads to a better choice of threshold. The hyperplane, obtained by determining the value of θ for which

$$\phi\left[\frac{\theta - \bar{w}'\bar{\mu}^1}{(\bar{w}'A^1\bar{w})^{1/2}} \right] \quad \text{and} \quad \phi\left[\frac{\bar{w}'\bar{\mu}^1 - \theta}{(\bar{w}'A^2\bar{w})^{1/2}} \right]$$

are equal, is H_5 in Fig. 2.

Another possible approximation consists of replacing the intraclass scatter matrix A in (5) by the identity matrix. This corresponds to maximizing the scatter of the projected points in the two classes pooled together, and is also equivalent to principal components analysis. This hyperplane, with $\theta = \frac{1}{2}\bar{w}'(\bar{\mu}^1 + \bar{\mu}^2)$, is H_6 in Fig. 2.

Trainable Categorizers

In applying the statistical algorithms of the preceding sections, all of the patterns in the training set were considered simultaneously to compute the weight vector. In a trainable categorizer, however, the patterns are

TABLE II

TRAINING A CATEGORIZER

Step	Pattern	$\bar{w}'\bar{x}-\theta$	w_1	w_2	θ	Increment?
1	$\bar{x}_1$	0.00	0.00	4.83	−1.00	
2	$\bar{x}_5$	−14.67	−4.00	2.00	−0.00	*
3	$\bar{x}_2$	− 1.66	−4.00	1.17	−1.00	*
4	$\bar{x}_6$	18.31	−4.00	1.17	−1.00	
5	$\bar{x}_3$	− 6.46	−1.55	3.17	−2.00	*
6	$\bar{x}_7$	5.96	−1.55	3.17	−2.00	
7	$\bar{x}_4$	12.14	−1.55	3.17	−2.00	
8	$\bar{x}_8$	2.01	−1.55	3.17	−2.00	
9	$\bar{x}_1$	17.31	−1.55	3.17	−2.00	
10	$\bar{x}_5$	− 4.77	−5.55	0.34	−1.00	*
11	$\bar{x}_2$	0.72	−5.55	0.34	−1.00	
12	$\bar{x}_6$	22.16	−5.55	0.34	−1.00	
13	$\bar{x}_3$	−11.92	−3.10	2.34	−2.00	*
14	$\bar{x}_7$	13.93	−3.10	2.34	−2.00	
15	$\bar{x}_4$	14.28	−3.10	2.34	−2.00	
16	$\bar{x}_8$	6.03	−3.10	2.34	−2.00	
17	$\bar{x}_1$	13.30	−3.10	2.34	−2.00	
18	$\bar{x}_5$	3.78	−3.10	2.34	−2.00	
19	$\bar{x}_2$	0.06	−3.10	2.34	−2.00	
20	$\bar{x}_6$	17.02	−3.10	2.34	−2.00	
21	$\bar{x}_3$	− 0.92	−0.65	4.34	−3.00	*
22	$\bar{x}_7$	0.34	−0.65	4.34	−3.00	
23	$\bar{x}_4$	13.27	−0.65	4.34	−3.00	
24	$\bar{x}_8$	− 1.32	−3.24	4.34	−2.00	*
25	$\bar{x}_1$	22.97	−3.24	4.34	−2.00	
26	$\bar{x}_5$	− 1.32	−7.24	1.51	−1.00	*
27	$\bar{x}_2$	− 0.25	−7.24	0.68	−2.00	*
28	$\bar{x}_6$	28.88	−7.24	0.68	−2.00	
29	$\bar{x}_3$	−14.38	−4.79	2.68	−3.00	*
30	$\bar{x}_7$	21.62	−4.79	2.68	−3.00	
31	$\bar{x}_4$	20.10	−4.79	2.68	−3.00	
32	$\bar{x}_8$	9.41	−4.79	2.68	−3.00	
33	$\bar{x}_1$	15.94	−4.79	2.68	−3.00	
34	$\bar{x}_5$	8.58	−4.79	2.68	−3.00	
35	$\bar{x}_2$	0.78	−4.79	2.68	−3.00	
36	$\bar{x}_6$	23.74	−4.79	2.68	−3.00	
37	$\bar{x}_3$	− 3.38	−2.34	4.68	−4.00	*
38	$\bar{x}_7$	8.08	−2.34	4.68	−4.00	
39	$\bar{x}_4$	19.09	−2.34	4.68	−4.00	
40	$\bar{x}_8$	2.06	−2.34	4.68	−4.00	
41	$\bar{x}_1$	26.60	−2.34	4.68	−4.00	
42	$\bar{x}_5$	− 7.88	−6.34	1.85	−3.00	*
43	$\bar{x}_2$	1.46	−6.34	1.85	−3.00	
44	$\bar{x}_6$	27.59	−6.34	1.85	−3.00	
45	$\bar{x}_3$	− 8.83	−3.89	3.85	−4.00	*
46	$\bar{x}_7$	15.99	−3.89	3.85	−4.00	
47	$\bar{x}_4$	21.23	−3.89	3.85	−4.00	
48	$\bar{x}_8$	6.08	−3.89	3.85	−4.00	
49	$\bar{x}_1$	22.60	−3.89	3.85	−4.00	
50	$\bar{x}_5$	0.66	−3.89	3.85	−4.00	
51	$\bar{x}_2$	0.80	−3.89	3.85	−4.00	
52	$\bar{x}_6$	22.46	−3.89	3.85	−4.00	
53	$\bar{x}_3$	2.17	−3.89	3.85	−4.00	
54	$\bar{x}_7$	15.99	−3.89	3.85	−4.00	
55	$\bar{x}_4$	21.23	−3.89	3.85	−4.00	
56	$\bar{x}_8$	6.08	−3.89	3.85	−4.00	

CLASS C^1:

$$\bar{x}_1 = \begin{pmatrix} 0.00 \\ 4.83 \end{pmatrix} \qquad \bar{x}_2 = \begin{pmatrix} 0.00 \\ -0.83 \end{pmatrix} \qquad \bar{x}_3 = \begin{pmatrix} 2.45 \\ 2.00 \end{pmatrix} \qquad \bar{x}_4 = \begin{pmatrix} -2.45 \\ 2.00 \end{pmatrix}$$

CLASS C^2:

$$-\bar{x}_5 = \begin{pmatrix} 4.00 \\ 2.83 \end{pmatrix} \qquad -\bar{x}_6 = \begin{pmatrix} 4.00 \\ -2.83 \end{pmatrix} \qquad -\bar{x}_7 = \begin{pmatrix} 5.41 \\ 0.00 \end{pmatrix} \qquad -\bar{x}_8 = \begin{pmatrix} 2.59 \\ 0.00 \end{pmatrix}$$

Note: The $\overline{X}_j$'s in class C^2 have been multiplied by -1 to simplify the computation by requiring a positive response for every pattern.

presented one at a time, and the weight vector is changed incrementally throughout the process. This mode of operation offers some advantage in implementing the algorithm in hardware, but the final error rates achievable by the two approaches appear to be substantially the same.

There are many algorithms which guarantee convergence to the optimal weight vector under various conditions.[30] One of the earliest for which a bound on the maximum number of steps required was obtained is the error correcting algorithm of perceptron fame.[37] Here, all the patterns in the training set are presented in sequence; when the training set is exhausted, the patterns are presented again in the same order. Thus, if there are N patterns available, the jth step involves the ith pattern, where $i=j$ modulo N. The weights change according to

$$\begin{aligned} \bar{w}_{j+1} &= \bar{w}_j + \bar{x}_j \quad \text{if } \bar{w}_j\bar{x}_j \le \theta \quad \text{and} \quad \bar{x}_j \in C^1 \\ &= \bar{w}_j - \bar{x}_j \quad \text{if } \bar{w}_j\bar{x}_j \ge \theta \quad \text{and} \quad \bar{x}_j \in C^2 \\ &= \bar{w}_j \qquad \text{otherwise.} \end{aligned} \tag{6}$$

The weight vector is changed only after a pattern has been misidentified. The initial vector $\bar{w}_0$ may be the null vector, or, better, a coarse approximation such as may be obtained with (1). The threshold may be derived by extending the dimensionality of the pattern space by one, and setting the corresponding component

of all the patterns equal to 1. The weight associated with this component is the required threshold.

The manner of convergence of the algorithm is shown on the two-dimensional problem in Table II. The final weight vector is H_7 in Fig. 2.

Since 1957, when a proof for the convergence of this algorithm was outlined by Rosenblatt, at least six or seven more or less independent proofs have been advanced. One derivation, and the corresponding bound on the number of steps, is given by Singleton[40] in terms of the pattern matrix B. The ith column of B is $\bar{x}_i$ if $\bar{x}_i$ belongs to C^1, and $-\bar{x}_i$ if it belongs to C^2.

The theorem states that if $\exists \bar{w} \exists B^t \cdot \bar{w} > 0$, then

$$\bar{w}_{k+1} = \bar{w}_k \quad \text{for } k \geq \frac{(\bar{w} \cdot \bar{w}) \max_i (B^t B)_{ii}}{[\min_i (B^t w)_i]^2} . \tag{7}$$

This means that if the categorization problem does have a linear solution, an incremental adaptation procedure will find it in a finite number of steps. The similarity to the achievement of feasible solutions in linear programming has been repeatedly pointed out.[36],[41]

Unfortunately, the upper bounds given in the literature for the number of adjustments needed, all require that at least one solution to the problem be known before the length of the training sequence can be estimated. In the example in Table II, the bound calculated from (7) is 284 steps, whereas convergence is actually reached in 45 steps.

Variations of the theorem deal with the effects of varying the order of presentation of the training patterns, of changing the total amount added to the weights depending on how close the response was to being right, of requiring a dead-zone between the pattern classes, and of imperfect components.[16],[19]–[22],[27],[28],[34]

Joseph,[21] for example, has proved that the fundamental convergence theorem holds even if the number of levels in each adaptive link is finite, i.e., if the storage elements are saturable. Low[27] has a similar demonstration for the case of nonuniform adaptation.

Perturbations in the final values of individual weights introduce errors. Hoff[20] has shown that the expected probability of errors, on patterns with a uniform distribution of ones and zeros, is roughly

$$p(\epsilon) = \frac{3}{4} \frac{\delta}{|\bar{w}|} \sqrt{n}$$

where δ is the average drift in the weights, $\bar{w}$ is the solution weight vector (before drifting), and n is the number of weights.

The convergence theorem, with all its variants and corollaries, applies only if a solution exists. It is not difficult to show that solutions exist if and only if there are no linear dependencies between input patterns, considered as vectors, in opposite classes. In other words, conflicts of the type seen in the example in Fig. 3 must not arise.

It is no trivial matter, however, to look at several thousand 100- or 500-dimensional vectors, and spot the linear dependencies. A number of procedures, some of which take advantage of the statistical distribution of ones and zeros in the input vectors, have been devised to carry out this operation, but the most common method for finding out if a problem is linearly separable is to simulate the linear categorizer on a computer and try to teach it the required classification.[14],[39],[40] If, after many presentations of the pattern, it has not been learned, then it is assumed that unwanted linear dependencies do occur. Several adapters have noticed the oscillatory behavior of the weight vector when presented with an insoluable task; this symptom of frustration provides a valuable clue as to when to stop training.[12],[23],[34]

It would be comforting to know that, if the problem is not completely solvable, the weights converge to the values guaranteeing a minimum number of mistakes among the training samples. This, however, is not necessarily the case; the algorithm can be stranded on a local optimum.

Chow[7] points out that the assumptions leading to the procedures specified by (2) and (6), statistical independence and linear separability, are mutually contradictory for binary patterns, except in trivial instances. To circumvent this difficulty, several gradient methods, which cover the gamut between the "little at a time" and the "all at once" approaches, have been developed.[10],[17],[25]

Nonlinear Categorizers

With the exception of a few special distributions, very little is known about approximating the optimum nonlinear boundaries in classification problems involving pattern vectors with numerical (as opposed to binary) correlated components. For Gaussian distributions, the optimal separating surface is easily shown to be a hyperquadric, specified by the following equations:[9]

$$(\bar{x} - \mu^1)' A^{1^{-1}} (\bar{x} - \bar{\mu}^1) + \log \det A^1 = (\bar{x} - \bar{\mu}^2)' A^{2^{-1}} (\bar{x} - \bar{\mu}^2) + \log \det A^2 \tag{8}$$

where A^k is the covariance matrix of class k and $\bar{\mu}^k$ is the mean vector of class k. This boundary is shown as H_8 in Fig. 2 for the distributions used to illustrate the linear methods.

In addition to Gaussian distributions, the hyperquadric is optimal for Pearson distributions type II and type VII; these cases have also been fully analyzed by Cooper. The equations for the separating surface for a pair of parametric distributions can, of course, always be obtained by setting the difference of the density functions equal to zero, but since, in a practical situation, the difficulty usually lies in estimating the pa-

rameters from the samples, this is really begging the problem.

In an n-dimensional binary classification task, every vertex of the n-dimensional hypercube represents a possible pattern. Thus, the general solution requires 2^n values for complete specification, as opposed to the n values for the linear separation. In practice, even the n^2 values required by considering all of the components pairwise without additional simplifying assumptions represents too much of a burden for implementation or simulation.

The probability distributions we must compare in order to decide the class assignment can be written as product expansions. In the following expressions, the single subscript refers to the components of the pattern vector:

$$P(\bar{x} \mid C^k) = P(x_1 \mid C^k) P(x_2 \mid x_1, C^k) \cdots P(x_j \mid x_{j-1}, x_{j-2}, \cdots, x_1, C^k) \cdots .$$

Each variable is conditioned upon all the variables appearing in the preceding factors. The product can be rewritten as a sum of weighted terms by taking logarithms and assigning a weight component to every possible sequence of ones and zeros in the partial pattern vector in each term. Thus,

$$\ln P(\bar{x} \mid C^k) = w_1 x_1 + w_2(1 - x_1) + w_3 x_1 x_2 + w_4(1 - x_1)x_2 + w_5 x_1(1 - x_2) + w_6(1 - x_1)(1 - x_2) \cdots .$$

The customary procedure is to neglect all but the second-order terms (each component conditioned on only one other component), and to select even among these only the most important pairs. In some cases, a natural ordering is available. For example, if the pattern is the binary matrix representation of a character, it is reasonable to let each x_{ij} depend only on its "nearest neighbors."[6] If, however, the binary components represent the results of arbitrary measurements upon the patterns, then a natural ordering is not available, and the most important pairs must be found by heuristic methods. "Chain" and "tree" representations based on performance criteria have been advocated by Chow.[8] An alternative is to let a trainable machine adapt its weights on inputs representing every pair in all four possible combinations, and to select the pairs with the largest weights for ultimate use.

A simple illustration of the effect of correlation among the pattern components is given in Fig. 4. Here, two patterns differ from a third by the same number of bits, but in one case the mismatching locations appear to be highly correlated, while in the other they are independently distributed.

The selection of correlated points in a pattern is closely related to the problem of measurement design or feature extraction, which is outside the scope of this paper.

Fig. 4. Correlations in binary patterns. The center pattern, which may be considered the "unknown," differs from each of the outside patterns ("templates") by 9 bit positions. The effect of the correlations among the mismatch bits must thus be taken into account for correct identification. Although this is an artificially constructed example, instances of such neighborhood correlations frequently occur in practice.

Multiclass Problems and Sequential Decisions

With m pattern classes, the number of discriminants (hyperplanes or other surfaces) necessary to separate the *means* of the classes may vary from $\log_2 m$ to $m-1$, depending on the location of the means. Fig. 5 gives examples of both extremes.

In general, the classes must be assigned to discriminants before the parameters characterizing the discriminants are determined from the training set. Let a "one" be associated with the positive side of each surface (as with the hyperplanes), and a "zero" with the negative side. Then, each sample is characterized by a string of ones and zeros, the code for that sample.

Even when all m discriminants are used for an m-class problem, the natural code of 1-out-of-m may not be the best. It is sometimes argued that a maximum distance code, where each discriminant attempts to separate several pairs of classes, would be superior. The question is which classes to lump together for each discriminant. In character recognition, for example, O's, C's, D's, G's, and Q's are occasionally assigned to a single hyperplane, with subsequent separation by more detailed criteria. Kiessling has proposed several heuristics which bear on the problem.[3],[24]

Once we consider coding the classes in this manner, we necessarily introduce another level in the decision process, and the distinction between "categorization" and "feature extraction" begins to blur. There also appears the possibility of sequential decision, where whether we look at the result of another measurement, and which other measurement, depends on the outcome of the previous measurements.

Fu, Chen, and Chien[5],[13] have examined in detail the economies in computation which may be realized through the application of sequential decision models in pattern recognition. The strategy is simple; the next measurement chosen is always the one which gives the most information about the class pair with the highest residual probability of error. The interrogation of measurements is halted when the estimated error probability reaches a preset threshold, or when all the measurements in a particular branch of the decision tree are exhausted. In order to apply the theory to practical problems, a great many assumptions must be satisfied. Nevertheless, small-scale experiments show promising results.

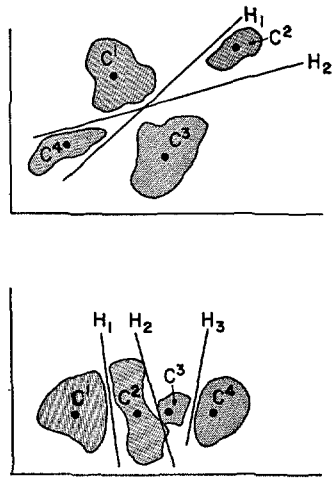

Fig. 5. Multiclass categorizers. The minimum number of hyperplanes necessary to partition a given number of classes depends on their disposition in hyperspace. The top diagram shows an example where $\log_2 4=2$ hyperplanes suffice. In the bottom diagram, where the classes are roughly colinear, $4-1=3$ hyperplanes are required. Of course, the difference between $\log_2 m$ and $m-1$ increases with the number of classes m.

A much simpler form of sequential decision, involving only two levels, has been successfully applied to the recognition of both English and Chinese characters.[4],[26]

Other Aspects of Pattern Recognition

Practical considerations of computer economics sometimes prevent the wholesale application of the methods mentioned above to real-life situations. The somewhat undignified and haphazard manipulation, invoked in such cases to render the problem amenable to orderly solution, is referred to variously as preprocessing, filtering or prefiltering, feature or measurement extraction, or dimensionality reduction. One distinction between these concepts and the ones mentioned earlier is that here the differences between the several classes of data under consideration are not necessarily taken into account in the selection of the parameters. Instead, the object is to provide a simplified, more economical description, which applies to all the samples. These matters are discussed by the author at greater length in a forthcoming state-of-the-art report in the Proceedings of the IEEE.

There have been several attempts to readjust parameters according to the output of a recognition system, and to track data sets with slowly changing characteristics. This forms the subject matter of unsupervised learning.[42]

The need for automatic methods to cluster samples according to their relative similarity first arose in numerical taxonomy, but applications for the resulting clustering techniques rapidly emerged in many areas of pattern recognition. Some simple clustering algorithms which can be used to simulate a species of self-organizing behavior are described in Ball.[2]

Applications

What are all these schemes good for? Only character recognition has received widespread commercial acceptance so far. Machines are gradually upgrading their reading ability from the simple magnetic symbols used on bank checks to the somewhat less stylized numerals on credit cards and other turnaround documents, and even to typescripts in more or less arbitrary typestyles. A really general purpose page reader is still to be developed, but the problem areas lie more in the realm of format than in character recognition per se. Several machines have already been marketed to read hand-printed numerals in well-specified surroundings.

Speech recognition is probably the next broad area to reach commercial maturity. Hopefully, the expanding vocabularies, decreasing sensitivity to variations between individual speakers, greater overall accuracy, and the capability of processing longer and longer segments of continuous speech will prove irresistibly attractive to system designers preoccupied with man–machine communications.

Other areas where much effort is being expended to develop viable systems include aerial and microphotograph processing; particle tracking in cloud, bubble, and spark chambers; seismic signal analysis for both geophysical exploration and explosion monitoring; electrocardiogram, electroencephalogram, and single-fiber recording in medicine and physiology; fingerprint analysis; and weather prediction.

Additional Sources of Information

New results in pattern recognition are presented every year at some 20 conventions, symposia, congresses, and workshops of national caliber. All of the major computer meetings devote one or more sessions to pattern recognition or signal processing; other papers are presented at the meetings of learned societies in statistics, automatic control, information theory, cybernetics, acoustics, communications, and in special application areas of physics, chemistry, biology, and medicine.

In the technical press, the IEEE publishes the largest number of papers in this field. In 1966, about four dozen papers were published in the IEEE Transactions on Electrical Computers, Information Theory, Systems Science and Cybernetics, Automatic Control, Bio-Medical Engineering, and Audio and Electroacoustics. Other publications which regularly devote space to pattern recognition are *Information and Control*, the *Journal* and *Communications of the Association for Computing Machinery*, and the *Computer Journal*.

A far larger number of papers is published annually in the proceedings of the conferences previously mentioned, as reports on government contracts, as company reports, and as dissertations at universities.

It is to be hoped that at least some of this work proves applicable in the field of audio and speech research.

Bibliography

[1] T. W. Anderson and R. Bahadur, "Classification into two multivariate normal distributions with different covariance matrices," *Ann. Math. Stat.*, vol. 33, pp. 422–431, 1962.

[2] G. H. Ball, "Data analysis in the social sciences: what about the details?" *Proc. Fall Joint Computer Conf.* (Las Vegas, Nev., December 1965), pp. 533–559.

[3] E. M. Braverman, "Experiments on machine learning to recognize visual patterns," translated from *Automation i Telemakhanika*, vol. 23, pp. 349–364, March 1962.

[4] R. G. Casey and G. Nagy, "Recognition of printed Chinese characters," *IEEE Trans. Electronic Computers*, vol. EC-15, pp. 91–101, February 1966.

[6] Y. T. Chien and K. S. Fu, "A modified sequential recognition machine using time-varying stopping boundaries," *IEEE Trans. Information Theory*, vol. IT-12, pp. 206–214, April 1966.

[6] C. K. Chow, "A recognition method using neighbour dependence," *IRE Trans. Electronic Computers*, vol. EC-11, October 1962.

[7] ——"Statistical independence and threshold functions," *IEEE Trans. Electronic Computers*, vol. EC-14, pp. 66–68, February 1965.

[8] —— "A class of nonlinear recognition procedures," *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-2, December 1966.

[9] P. W. Cooper, "Hyperplanes, hyperspheres, and hyperquadrics as decision boundaries," in *Computer and Information Science*. Washington: Spartan Books, 1964, pp. 111–139.

[10] R. O. Duda and R. C. Singleton, "Training a threshold logic unit with imperfectly classified patterns," *Proc. Western Joint Computer Conf.* (Los Angeles, Calif., August 1964).

[11] R. O. Duda and H. Fossum, "Pattern classification by iteratively determined linear and piecewise linear discriminant functions," *IEEE Trans. Electronic Computers*, vol. EC-15, pp. 221–232, April 1966.

[12] B. Efron, "The perceptron correction procedure in nonseparable situations," Stanford Research Institute, Menlo Park, Calif., *Appl. Phys. Lab. Research Note*, August 1963.

[13] K. S. Fu and C. H. Chen, "A sequential decision approach to problems in pattern recognition and learning," *Proc. 3rd Symp. on Discrete Adaptive Processes* (Chicago, Ill., October 1964).

[14] C. A. Gaston, "A simple test for linear separability," *IEEE Trans. Electronic Computers*, vol. EC-12, April 1963.

[15] I. J. Good, "The estimation of probabilities," M. I. T. Press, Cambridge, Mass., Research Monograph 30, 1965.

[16] J. S. Griffin, J. H. King and C. J. Tunis, "A pattern identification system using linear decision functions," *IBM Systems J.*, vol. 2, pp. 248–267, December 1963.

[17] G. F. Groner, "Statistical analysis of adaptive linear classifiers," Stanford Electronics Labs., Stanford, Calif., Tech. Rept. 6761, April 1964.

[18] M. R. Hestenes and E. Stiefel, "Method of conjugate gradients for solving linear systems," *J. Research NBS*, vol. 49, pp. 409–436, 1952.

[19] W. H. Highleyman, "Linear decision functions with application to pattern recognition," *Proc. IRE*, vol. 50, pp. 1501–1514, June 1962.

[20] M. E. Hoff, Jr., "Learning phenomena in networks of adaptive switching circuits," Stanford Electronics Labs., Stanford, Calif., Tech. Rept. 1554-1, July 1962.

[21] R. D. Joseph, "On predicting perceptron performance," *1960 IRE Nat. Conv. Rec.*, pt. 2.

[22] C. Kesler, "Analysis and simulations of a nerve cell model," Cornell University, Ithaca, N. Y., in Cognitive Systems Research Program Rept. 2, May 1961.

[23] ——, "Further studies of reinforcement procedures and related problems," in *Collected Technical Papers*, vol. 2, Cognitive Systems Research Program, Cornell University, Ithaca, N. Y., pp. 16–64, July 1963.

[24] C. Kiessling, "The generation of linearly separable codes for adaptive threshold networks," Masters thesis, Dept. of Elec. Engrg., Cornell University, Ithaca, N. Y., 1965.

[25] J. S. Koford, "Adaptive pattern dichotomization," Stanford Electronics Labs., Stanford, Calif., Tech. Rept. 6201-1, April 1964.

[26] C. N. Liu and G. L. Shelton, Jr., "An experimental investigation of a mixed-font print recognition system," *IEEE Trans. Electronic Computers*, vol. EC-15, pp. 916–925, December 1966.

[27] P. R. Low, "Influence of component imperfections on trainable systems," *1963 Proc. WESCON*, Conf., paper 6-3.

[28] C. H. Mays, "Effect of adaptation parameters on convergence time and tolerance for adaptive threshold elements," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 465–468, August 1964.

[29] M. Minsky, "Steps towards artificial intelligence," *Proc. IRE*, pp. 8–30, January 1961.

[30] N. J. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.

[31] R. A. Penrose, "Generalized inverse for matrices," *Proc. Cambridge Phil. Soc.*, vol. 51, pp. 406–413, 1955.

[32] D. W. Peterson, "Discriminant functions: properties, classes, and computational techniques," Stanford Electronics Labs., Stanford, Calif., Tech. Rept. 6761-2, April 1965.

[33] J. Raviv and D. N. Streeter, "Linear methods for biological data processing," IBM Research Rept. RC-1577, Yorktown Heights, N. Y., December 1965.

[34] W. C. Ridgeway, III, "An adaptive logic system with generalizing properties," Stanford Electronics Labs., Stanford, Calif., Tech. Rept. 1556-1, April 1962.

[35] C. A. Rosen and D. J. Hall, "A pattern recognition experiment with near optimum results," *IEEE Trans. Electronic Computers*, vol. EC-15, pp. 666–667, August 1966.

[36] J. B. Rosen, "Pattern separation by convex programming," *J. Math. Anal. and Appl.*, vol. 10, pp. 123–134, 1965.

[37] F. Rosenblatt, "The perceptron, a perceiving and recognizing automaton," in Cornell Aeronautical Lab., Buffalo, N. Y., Rept. 85-460-1, January 1957.

[38] O. G. Selfridge and U. Neisser, "Pattern recognition by machine," *Scientific American*, pp. 60–68, August 1960.

[39] C. L. Sheng, "A method for testing and realization of threshold functions," *IEEE Trans. Electronic Computers*, vol. EC-13, June 1964.

[40] R. C. Singleton, "A test for linear separability as applied to self-organizing machines," in *Self-Organizing Systems*. Washington: Spartan Books, 1962.

[41] F. W. Smith, "Automatic HF signal classification by method of moments," Sylvania Electronics Systems, Rept. EDL-M1031, March 1967.

[42] J. Spragins, "Learning without a teacher," *IEEE Trans. Information Theory*, vol. IT-12, pp. 223–229, April 1966.

[43] S. S. Wilks, *Mathematical Statistics*. New York: Wiley, 1962.

George Nagy (M'66) was born in Budapest, Hungary, on July 7, 1937. He received the B. Engineering degree in engineering physics in 1959, and the M. Engineering degree in electrical engineering in 1960 from McGill University, Montreal, Canada, and the Ph.D. degree from Cornell University, Ithaca, N. Y., in 1962.

From 1962 to 1963 he was a Research Associate at the Cognitive Systems Research Program of Cornell University, where he worked with F. Rosenblatt on the audio-perception Tobermory. Since 1963, he has been engaged in research on adaptive pattern recognition systems at the IBM Watson Research Center in Yorktown Heights, N. Y. He has published papers on Chinese character recognition, adaptive devices and systems, self-corrective character recognition, analog matrix multipliers, and read-only memory devices. At present he is on leave of absence from the IBM Corporation at the University of Montreal, Montreal Canada, where he is attempting to apply pattern recognition techniques to neurophysiological data.

Dr. Nagy is on the Pattern Recognition Committee of the IEEE Group on Electronic Computers.