H. J. Greenberg
A. G. Konheim

Linear and Nonlinear Methods in Pattern Classification

Abstract: The problem of pattern classification has two highly interactive aspects: (1) the selection of numerical measurements to 'represent' the patterns, and (2) the specification of an algorithm to identify patterns, based upon the numerical values of these measurements. The present paper presents the mathematical framework for one attack upon these problems and gives results obtained in some experiments in character recognition.

1. Introduction

In July, 1962, one of the authors* examined two closely related signal adaptive networks, the Perceptron of F. Rosenblatt[1,2] and a character recognition machine (PAPA) proposed by A. Gamba,[3] and pointed out their relationship to systems of linear inequalities. For an associated "linear" model, necessary and sufficient conditions for the convergence of the so-called training procedure were established. Subsequently these ideas were developed by the authors and applied to a number of problems in character recognition. Independently, other investigators (notably Widrow[4] and Griffin, King and Tunis[5]) considered the development of various realizations of linear adaptive networks.

The purpose of the present paper is to present the mathematical framework for one attack upon these problems and to summarize the results obtained in some experiments in character recognition. In Sections 2 and 3 the problem is stated and certain mathematical preliminaries are covered. In Section 4 the special case of linear separability is treated and in Section 5 it is shown how the procedure can be modified to obtain nonlinear separability by polynomials. In Section 6 the results of certain experiments in character recognition are discussed; these center on the linearly separable case.

2. Statement of the problem

The problem of pattern classification has two highly interactive aspects: (1) the selection of numerical measurements to 'represent' the patterns; and (2) the specification of an algorithm to identify a pattern based upon the numerical values of these measurements. Our investigations have been addressed to the second aspect but, as will come out in Section 3, the solution reached also provides information and guidance in the selection of the relevant measurements.

The space of patterns (also referred to as input signals to the classification system) will be denoted by Ω; ω will denote a generic element of Ω. The space Ω is assumed to have been partitioned a priori into subsets $\Omega_1, \Omega_2, \cdots, \Omega_m$ corresponding to the distinct responses required of the classification system. For example, in alphanumeric character recognition, there may be 62 classes (26 upper case letters, 26 lower case letters and 10 numerals). We indicate the fact that ω belongs to Ω_i by writing $\omega \epsilon \Omega_i$.

Each classification system essentially consists of two components: One is a *transducer* in which a sequence of, say, n numerical-valued measurements $X_1, X_2, \cdots, X_n$ is made upon each input signal ω. We will write $X_i(\omega) = x_i$ to signify that the i^{th} measurement upon the signal ω resulted in the numerical quantity x_i. With each input signal ω we associate its vector $\mathbf{X} = \mathbf{X}(\omega) = (x_1, x_2, \cdots, x_n)$ of measurements. The other component is a *processor* which classifies ω by observing the results $\mathbf{X}(\omega)$ of the measurements upon ω.

The transducer and processor can be thought of as transformations which are applied to the original data Ω; the transducer first maps each point ω of Ω into a point $\mathbf{x}$

* Konheim, A. G., "A Note on Adaptive Networks," IBM Conference on Non-Numeric Processing, Thomas J. Watson Research Center, July 16, 1962.

of the n-dimensional vector space R^n. The processor then partitions R^n into disjoint sets $A_1, A_2, \cdots, A_m$ and if the results of the measurements upon ω lie in A_i, $\mathbf{X}(\omega) \epsilon A_i$, the processor makes the decision $\omega \epsilon \Omega_i$.

For a given set of measurements $X_1, X_2, \cdots, X_n$ a processor exists which achieves the proper identification of points in Ω provided that there are sufficiently many measurements to distinguish between the sets $\{\Omega_i\}$, i.e., $\mathbf{X}(\omega) \neq \mathbf{X}(\omega')$ if ω and ω' belong to different subsets $\{\Omega_k\}$. An approach as general as this is rarely taken because of the technical problems inherent in the realization of the processor. The usual procedure is to specify the nature of the processor in advance, i.e., the types of partitions $\{A_i\}$ of R^n, and then to consider only problems which can be solved within this class of partitions. In this paper we shall study a class of partitions $\{\mathfrak{A}_k: k = 1, 2, \cdots\}$ of increasing complexity. The index k refers to the degree of certain polynomials employed in the construction of the partition. As k increases, the partitions $\{\mathfrak{A}_k\}$ become capable of solving successively more complicated problems. Furthermore, (1) The partitions $\{\mathfrak{A}_k\}$ are sufficient in all cases where Ω is a finite set and the measurements $X_1, X_2, \cdots, X_n$ distinguish between the sets $\{\Omega_k\}$. (We consider the mapping $\mathbf{X}$ of Ω into R^n defined by $\omega \rightarrow \mathbf{X}(\omega) = [X_1(\omega), X_2(\omega), \cdots, X_n(\omega)]$ and require that the sets $\{\mathbf{X}(\Omega_k)\}$ be disjoint.) (2) The partitions $\{\mathfrak{A}_k\}$ may be "essentially" realized by threshold devices.

3. Preliminaries

R^n will denote the set of all *vectors* $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ with real components. R^n is a vector space with vector addition and scalar multiplication being defined as usual by

$$\mathbf{x} + \mathbf{y} = (x_1, x_2, \cdots, x_n) + (y_1, y_2, \cdots, y_n)$$
$$= (x_1 + y_1, x_2 + y_2, \cdots, x_n + y_n),$$
$$\alpha \mathbf{x} = \alpha(x_1, x_2, \cdots, x_n) = (\alpha x_1, \alpha x_2, \cdots, \alpha x_n).$$

The *scalar* (or inner) product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is defined by $(\mathbf{x}, \mathbf{y}) = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$, and the *norm* of $\mathbf{x}$ by $||\mathbf{x}|| = (\mathbf{x}, \mathbf{x})^{1/2}$. We shall need in the sequel *Schwarz's inequality*

$$|(\mathbf{x}, \mathbf{y})| \leq ||\mathbf{x}||\, ||\mathbf{y}||.$$

A *linear functional* on R^n is a real-valued function L defined on R^n which satisfies $L(\alpha \mathbf{x} + \beta \mathbf{y}) = \alpha L(\mathbf{x}) + \beta L(\mathbf{y})$, where α and β are real numbers. Each such L admits the following simple representation: there is a vector $\mathbf{v}$ (depending upon L) such that $L(\mathbf{x}) = (\mathbf{v}, \mathbf{x})$.

Let $A = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_p\}$ and $B = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_q\}$ be two finite subsets of R^n. The pair (A, B) is said to be *linearly separable* if there exists a linear functional L such that

$$\max_{1 \leq i \leq q} L(\mathbf{b}_i) < \min_{1 \leq i \leq p} L(\mathbf{a}_i). \tag{1}$$

If c is any real number satisfying

$$\max_{1 \leq i \leq q} L(\mathbf{b}_i) < c < \min_{1 \leq i \leq p} L(\mathbf{a}_i), \tag{2}$$

then the sign of $L(\mathbf{x}) - c$ serves to identify the location of any point $\mathbf{x}$ belonging to $A \cup B$. (The set $A \cup B$ consists of all points belonging to either A or B.) The set of points which satisfy $L(\mathbf{x}) - c = 0$,

$$HP(L, c) = \{\mathbf{x} : L(\mathbf{x}) - c = 0\},$$

constitutes a hyperplane in R^n. This hyperplane $HP(L, c)$ is the boundary of the two half-spaces

$$HP(L, c)^+ = \{\mathbf{x} : L(\mathbf{x}) - c > 0\},$$
$$HP(L, c)^- = \{\mathbf{x} : L(\mathbf{x}) - c < 0\}.$$

According to Eqs. (1) and (2) the points of A lie in $HP(L, c)^+$ and the points of B in $HP(L, c)^-$; that is,

$$A \subset HP(L, c)^+,$$
$$B \subset HP(L, c)^-.$$

We now turn to two questions:

1. What conditions on (A, B) insure that a separating hyperplane exists?

2. If (A, B) are linearly separable, how can a separating hyperplane be found?

We first consider Question 1. A set of points in R^n, K (say) is called a *convex set* if whenever $\mathbf{x}$ and $\mathbf{y}$ are in K then so are all points on the "line segment" joining $\mathbf{x}$ and $\mathbf{y}$. These are the points

$$\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}, \tag{3}$$

where $0 \leq \lambda \leq 1$.

The points given in (3) are called convex combinations of $\mathbf{x}$ and $\mathbf{y}$. More generally, if $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m$ are any m points in R^n then all points of the form

$$\lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \cdots + \lambda_m \mathbf{x}_m \qquad \left(\lambda_i \geq 0,\ 1 \leq i \leq m,\ \sum_{i=1}^{m} \lambda_i = 1\right) \tag{4}$$

are called convex combinations of $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m$.

If $U = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$ is a set of points in R^n there exists a smallest convex set containing U; this set, called the *convex hull of* U and denoted by co (U), consists of all points of the form given by (4).

The connection between convexity and linear separation is provided by the following very classical statement:

• *Theorem*

*If A and B are finite sets in R^n, then (A, B) is linearly separable if and only if co $(A) \cap$ co $(B) = \phi$.**

In addressing ourselves to Question, 2 it will be to our advantage to reformulate the condition of Eq. (1) before proceeding to the actual construction of the required separating functional. Let $A = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_p\}$ and $B = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_q\}$ be two finite sets in R^n with co $(A) \cap$ co $(B) = \phi$. There exists therefore a vector $\mathbf{v} \epsilon R^n$ and a real number c such that

$$(\mathbf{v}, \mathbf{b}_j) < c < (\mathbf{v}, \mathbf{a}_i), \tag{5}$$

where $1 \leq i \leq p, 1 \leq j \leq q$.

$$\text{Let } \mathbf{v} = (v_1, v_2, \cdots, v_n), \tag{6}$$

$$\mathbf{a}_i = (a_{i1}, a_{i2}, \cdots, a_{in}), \tag{7}$$

where $1 \leq i \leq p$, and

$$\mathbf{b}_j = (b_{j1}, b_{j2}, \cdots, b_{jn}), \tag{8}$$

where $1 \leq j \leq q$.

It is sometimes convenient to imbed the above n-dimensional setup into R^{n+1} by defining

$$\mathbf{w} = (v_1, v_2, \cdots, v_n, -c), \tag{9}$$

$$\mathbf{a}_i^* = (a_{i1}, a_{i2}, \cdots, a_{in}, 1), \tag{10}$$

where $1 \leq i \leq p$, and

$$\mathbf{b}_j^* = (b_{j1}, b_{j2}, \cdots, b_{jn}, 1), \tag{11}$$

where $1 \leq j \leq q$.

Then, by Equation (5),

$$(\mathbf{w}, \mathbf{b}_j^*) < 0 < (\mathbf{w}, \mathbf{a}_i^*), \tag{12}$$

where $1 \leq i \leq p, 1 \leq j \leq q$.

Conversely if there exists a vector $\mathbf{w} \epsilon \mathbf{R}^{n+1}$ which satisfies Eq. (12) (with $\mathbf{a}_i^*$, $\mathbf{b}_j^*$ being related to $\mathbf{a}_i$, $\mathbf{b}_j$ by Equations (7), (8), (10), and (11)), then the system of Eq. (5) admits a solution $\mathbf{v}$, with $\mathbf{v}$ and $\mathbf{w}$ being related by Eqs. (6) and (9).

Next we observe that since A and B are finite sets there must exist a $\theta > 0$ such that

$$(\mathbf{w}, \mathbf{b}_j^*) < -\theta < 0 < \theta < (\mathbf{w}, \mathbf{a}_i^*) \tag{13}$$

with $1 \leq i \leq p, 1 \leq j \leq q$.

But if (13) has a solution for *some* $\theta > 0$, it has, by homogeneity, a solution for *every* $\theta > 0$. The algorithm for determining the $\mathbf{v}$ and c in Eq. (5) is applied to the system of linear inequalities, Eq. (13), and our preliminary transformations have just shown that these two systems of inequalities are equivalent. In fact, if

$$\mathbf{w} = (w_1, w_2, \cdots, w_n, w_{n+1}) \tag{14}$$

is a solution to Eq. (13), then by setting

$$\mathbf{v} = (w_1, w_2, \cdots, w_n) \tag{15}$$

we will have

$$(\mathbf{v}, \mathbf{b}_j) < -\theta - w_{n+1} < \theta - w_{n+1} < (\mathbf{v}, \mathbf{a}_i), \tag{16}$$

where $1 \leq i \leq p, 1 \leq j \leq q$.

Let $\mathfrak{T} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots\}$ be any sequence of vectors chosen from the set $A^* \cup B^*$ where $A^* = \{\mathbf{a}_1^*, \mathbf{a}_2^*, \cdots, \mathbf{a}_p^*\}$ and $B^* = \{\mathbf{b}_1^*, \mathbf{b}_2^*, \cdots, \mathbf{b}_q^*\}$. In the literature of adaptive networks the set $\mathfrak{T}$ is referred to as a *training set.* A fixed $\theta > 0$ is chosen and the sequence of *weight vectors* $\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \cdots$ is defined inductively as follows:

(1) $\mathbf{w}_0$ is an arbitrary element of R^{n+1}.

$$(2)\ \mathbf{w}_n = \begin{cases} \mathbf{w}_{n-1} + \mathbf{x}_n & \text{if } (\mathbf{w}_{n-1}, \mathbf{x}_n) \leq \theta \text{ and } \mathbf{x}_n \epsilon A^* \\ \mathbf{w}_{n-1} & \text{if } (\mathbf{w}_{n-1}, \mathbf{x}_n) > \theta \text{ and } \mathbf{x}_n \epsilon A^* \\ \mathbf{w}_{n-1} - \mathbf{x}_n & \text{if } (\mathbf{w}_{n-1}, \mathbf{x}_n) \geq -\theta \text{ and } \mathbf{x}_n \epsilon B^* \\ \mathbf{w}_{n-1} & \text{if } (\mathbf{w}_{n-1}, \mathbf{x}_n) < -\theta \text{ and } \mathbf{x}_n \epsilon B^* \end{cases}$$

$$n = 1, 2, \cdots$$

• *Theorem*

The sequence $\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \cdots$ converges. There is an integer N (depending upon A^*, B^*, θ and $\mathbf{w}_0$) such that $\mathbf{w}_N = \mathbf{w}_{N+1} = \cdots$. If $\mathfrak{T}$ has the property that each element of $A^* \cup B^*$ occurs infinitely many times, then $\mathbf{w}_N$ is a solution of Eq. (13).

• *Proof*

This theorem has a very interesting history. It was first conjectured and proved by Rosenblatt in Refs. 1 and 3, where the components of the weight vectors $\{\mathbf{w}_i\}$ were the amplification factors of the association units (A-units) in Rosenblatt's simple three layer series-coupled Perceptron. The algorithm was called by Rosenblatt the *error correction procedure* since the amplification factors were changed, $\mathbf{w}_n \rightarrow \mathbf{w}_{n+1}$, during the training period only if the present amplification factors given by $\mathbf{w}_n$ failed to correctly identify the n^{th} input $\mathbf{x}_n$ of the training set. This theorem has since been rediscovered by many workers in this field. The proof given here is essentially due to A. Novikoff.[6]

We define the vectors $\{\mathbf{y}_n\}$ by

$$\mathbf{y}_n = \begin{cases} \mathbf{x}_n & \text{if } \mathbf{x}_n \epsilon A^* \\ -\mathbf{x}_n & \text{if } \mathbf{x}_n \epsilon B^*. \end{cases}$$

If no integer N exists such that $\mathbf{w}_N = \mathbf{w}_{N+1} = \cdots$ then

* co(A) ∩ co(B) consists of all points belonging to both co(A) and co(B). The statement co(A) ∩ co(B) = ϕ means that co(A) and co(B) have no points in common.

there must be a sequence $i_1, i_2, \cdots, i_k, \cdots$ such that

$$\mathbf{w}_{i_k} = \mathbf{w}_{i_{k-1}} + \mathbf{y}_{i_k},$$

where $k = 1, 2, \cdots$, and $i_0 = 0$ by convention. We have

$$||\mathbf{w}_{i_k}||^2 = ||\mathbf{w}_{i_{k-1}}||^2 + ||\mathbf{y}_{i_k}||^2 + 2(\mathbf{w}_{i_{k-1}}, \mathbf{y}_{i_k})$$

$$||\mathbf{w}_{i_k}||^2 \leq ||\mathbf{w}_{i_{k-1}}||^2 + M + 2\theta,$$

where $M = \max_n ||\mathbf{y}_n||^2 < \infty$. It follows that

$$||\mathbf{w}_{i_k}|| \leq Ck^{1/2}, \quad k = 1, 2, \cdots \tag{17}$$

On the other hand Eq. (13) has a solution $\mathbf{w}$ and hence

$$(\mathbf{w}, \mathbf{w}_{i_k}) = (\mathbf{w}, \mathbf{w}_0) + \sum_{j=1}^{k} (\mathbf{w}, \mathbf{y}_{i_j})$$

$$\geq (\mathbf{w}, \mathbf{w}_0) + k\theta \geq Dk, \tag{18}$$

where $k = 1, 2, \cdots$

By Schwarz's inequality $(\mathbf{w}, \mathbf{w}_{i_k}) \leq ||\mathbf{w}||\ ||\mathbf{w}_{i_k}||$ and hence, Eqs. (17) and (18) yield

$$Dk \leq ||\mathbf{w}||\ k^{1/2},$$

which cannot hold for sufficiently large k.

If $\mathfrak{T}$ has the property that each element of $A^* \cup B^*$ appears in $\mathfrak{T}$ infinitely many times then the limiting vector $\mathbf{w}_N$ must be a solution of Eq. (13) for $\mathbf{w}_N$ satisfies

$$(\mathbf{w}_N, \mathbf{x}_m)\begin{cases} > \theta & \text{if } \mathbf{x}_m \in A^* \\ < -\theta & \text{if } \mathbf{x}_m \in B^* \end{cases}$$

$$m = N + 1, N + 2, \cdots,$$

and each element of $A^* \cup B^*$ appears in the sequence $\mathbf{x}_{N+1}, \mathbf{x}_{N+2}, \cdots$.

• *Example 1*

$n = p = q = 2$, $A = \{\mathbf{a}_1 = (0, 0), \mathbf{a}_2 = (1, 0)\}$, $B = \{\mathbf{b}_1 = (1, 1), \mathbf{b}_2 = (0, 1)\}$, $\theta = 1$, $\mathbf{w}_0 = (0, 0, 0)$, and $\mathfrak{T} = \{\mathbf{a}_1^*, \mathbf{a}_2^*, \mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{a}_1^*, \mathbf{a}_2^*, \mathbf{b}_1^*, \mathbf{b}_2^*, \cdots\}$.

$\mathbf{w}_0 = (0, 0, 0)$

$\mathbf{w}_1 = (0, 0, 1) = \mathbf{w}_0 + \mathbf{a}_1^*$

$\mathbf{w}_2 = (1, 0, 2) = \mathbf{w}_1 + \mathbf{a}_1^*$

$\mathbf{w}_3 = (0, -1, 1) = \mathbf{w}_2 - \mathbf{b}_1^*$

$\mathbf{w}_4 = (0, -2, 0) = \mathbf{w}_3 - \mathbf{b}_2^*$

$\mathbf{w}_5 = (0, -2, 1) = \mathbf{w}_4 + \mathbf{a}_1^*$

$\mathbf{w}_6 = (1, -2, 2) = \mathbf{w}_5 + \mathbf{a}_2^*$

$\mathbf{w}_7 = (0, -3, 1) = \mathbf{w}_6 - \mathbf{b}_1^*$

$\mathbf{w}_8 = (0, -3, 1) = \mathbf{w}_7$

$\mathbf{w}_9 = (0, -3, 2) = \mathbf{w}_8 + \mathbf{a}_1^*$

$\mathbf{w}_{10} = (0, -3, 2) = \mathbf{w}_9$

$\mathbf{w}_{11} = (-1, -4, 1) = \mathbf{w}_{10} - \mathbf{b}_1^*$

$\mathbf{w}_{12} = (-1, -4, 1) = \mathbf{w}_{11}$

$\mathbf{w}_{13} = (-1, -4, 2) = \mathbf{w}_{12} + \mathbf{a}_1^*$

$\mathbf{w}_{14} = (0, -4, 3) = \mathbf{w}_{13} + \mathbf{a}_2^*$

$\mathbf{w}_{15} = (-1, -5, 2) = \mathbf{w}_{14} - \mathbf{b}_1^*$

$\mathbf{w}_{16} = (-1, -5, 2) = \mathbf{w}_{15}$

$\mathbf{w}_{17} = (-1, -5, 2) = \mathbf{w}_{16}$

$\mathbf{w}_{18} = (0, -5, 3) = \mathbf{w}_{17} + \mathbf{a}_2^*$

$\mathbf{w}_{19} = (0, -5, 3) = \mathbf{w}_{18}$

$\mathbf{w}_{20} = (0, -5, 3) = \mathbf{w}_{19}$

$\mathbf{w}_{21} = (0, -5, 3) = \mathbf{w}_{20}$

$\mathbf{w}_{22} = (0, -5, 3) = \mathbf{w}_{21}$

$\cdots$

Hence if $\mathbf{v} = (0, -5)$ then

$$(\mathbf{v}, \mathbf{b}_i) < c < (\mathbf{v}, \mathbf{a}_i),$$

where $1 \leq i \leq 2$, $1 \leq j \leq 2$,

for any c with $-5 < c < 0$. We note that

$$\mathbf{w}_{22} = 4\mathbf{a}_1 + 4\mathbf{a}_2 - 4\mathbf{b}_1 - \mathbf{b}_2.$$

Remarks. This simple example shows how the algorithm for finding a separating linear functional may, in addition, provide information about the significance of the data being used. This information is of two kinds. First, it can indicate *which patterns* in the training set are relatively more significant for the purpose of separating the classes. Information of this last kind is contained in the representation of the final vector $\mathbf{v}$ as a linear combination of the patterns to be separated, i.e.

$$\mathbf{v} = (0, -5) = 4\mathbf{a}_1 + 4\mathbf{a}_2 - 4\mathbf{b}_1 - \mathbf{b}_2.$$

From the coefficients we see that $\mathbf{b}_2$, having the numerical coefficient 1, is weighted less than the other three patterns, which have a coefficient of 4. This suggests that $\mathbf{b}_2$ might be unimportant for distinguishing class A from class B. Indeed, if $\mathbf{b}_2$ is dropped entirely, and the process is repeated for just the three patterns $\mathbf{a}_1$ and $\mathbf{a}_2$ in A, and $\mathbf{b}_1$ in B, we are led to the same solution vector $\mathbf{w}$ as before (it is a coincidence that we get the identical vector) by employing the new combination

$$\mathbf{w} = (0, -5, 3) = 3\mathbf{a}_1^* + 5\mathbf{a}_2^* - 5\mathbf{b}_1^*.$$

Note that even without training on $\mathbf{b}_2^*$, the functional $\mathbf{w}$ in this case will "correctly" classify this pattern, since $\mathbf{w} \cdot \mathbf{b}_2^* = -2$ which is < -1, as required for identification as a member of class B.

The second kind of information contained in the separating functional is concerned with the *significance of the measurements* or bits used to represent the patterns. The final vector is $\mathbf{v} = (0, -5)$. It follows that the first bit or measurement or component of $\mathbf{v}$ is of no importance in the classification and can be dropped. Thus, to distinguish class A from class B we need only to look at the second component. While this is clear by inspection in this simple example, it must be remembered that where many bits or components and many patterns or points are involved, human observation is not adequate to the task. Generally, where the components of $\mathbf{v}$ are numerically small we would try dropping the corresponding measurements as being relatively unimportant for classification purposes.

4. Linear separability in pattern classification

Let the space of input signals Ω be partitioned into the m subsets $\Omega_1, \Omega_2, \cdots, \Omega_m$. The set of measurements $X_1, X_2, \cdots, X_n$ maps the set Ω_i into the subset A_i of R^n according to

$$\omega \rightarrow \mathbf{X}(\omega) = (X_1(\omega), X_2(\omega), \cdots, X_n(\omega)).$$

Linear separability in pattern classification refers to attempts to classify the patterns in Ω by constructing linear boundaries between the sets $A_1, A_2, \cdots, A_m$. Two possible procedures for such a classification are discussed next.

- *Procedure 1*

Construct m linear functionals $L_1, L_2, \cdots, L_m$. The linear functional L_i is to separate the pair of sets

$$(A_i, \bigcup_{\substack{j \\ j \neq i}} A_j), \quad \text{i.e.}$$

$$L_i(\mathbf{b}) < c_i < L_i(\mathbf{a}) (\mathbf{b} \in \bigcup_{\substack{j \\ j \neq i}} A_j, \mathbf{a} \in A_i),$$

where $\bigcup_{\substack{j \\ j \neq i}} A_j$ denotes all points in $\mathbf{X}(\Omega)$ excluding those of A_i. The appropriate convexity condition guaranteeing the existence of such a linear functional is

$$\text{co}\,(A_i) \cap \text{co}\,(\bigcup_{\substack{j \\ j \neq i}} A_j) = \phi.$$

The recognition procedure consists of evaluating the numerical quantities

$$L_i(\mathbf{x}(\omega)) - c_i = \delta_i(\omega), \text{ where } 1 \leq i \leq m.$$

For precisely one index, say $i = i_0$, will $\mathrm{x}_{i_0}(\omega) > 0$ and in this case the decision $\omega \in \Omega_{i_0}$ is made.

- *Procedure 2: Class-pair separation*

In Procedure 1, above, the patterns were classified by constructing one linear functional for each desired response. The functional L_i separated the i^{th} class from all of the remaining classes. In class-pair separation $m(m-1)/2$ linear functionals $L_{ij} (1 \leq i \leq j \leq m)$ are constructed. The functional L_{ij} serves only to distinguish between the classes A_i and A_j. We choose L_{ij} so that

$$L_{ij}(\mathbf{b}) < c_{ij} < L_{ij}(\mathbf{a}) (\mathbf{a} \in A_i, \mathbf{b} \in A_j).$$

Such a construction is possible provided that

$$\text{co}\,(A_i) \cap \text{co}\,(A_j) = \phi,$$

which expresses the convexity requirements of the second procedure. To identify the location of the point ω we note that if $\omega \in \Omega_j$ then

$$L_{ij}(x(\omega)) - c_{ij} < 0, \quad \text{where} \quad 1 \leq i \leq j,$$

$$L_{jk}(x(\omega)) - c_{jk} > 0, \quad \text{where} \quad j \leq k \leq m.$$

The experimental results reported upon in Section 5 relate to the first of these two procedures. It should be clear that the convexity requirements of the first procedure are more stringent than those of the second procedure.

Before considering more general forms of separation we should point out the connection between the Bayes maximum likelihood procedure and linear separability. The space Ω of input signals is considered to be a probability space with a probability measure Pr defined on a suitable σ-field of subsets of Ω. The measurements $\{X_i\}$ are assumed to be random variables taking on the values 0 and 1. The conditional probabilities $p_{ij} (1 \leq i \leq n, 1 \leq j \leq m)$ are defined by

$$p_{ij} = \Pr\{\omega : X_i(\omega) = 1/\Omega_j\}.$$

If the random variables are independent then

$$\Pr\{\omega : \mathbf{x}(\omega) = \mathbf{x}/\Omega_j\} = \prod_{i=1}^{n} p_{ij}^{x_i}(1 - p_{ij})^{1-x_i}$$

$$(= p(\mathbf{x}/\Omega_j)).$$

The maximum-likelihood procedure is to decide $\omega \in \Omega_j$ provided

$$\Pr\{\Omega_j\} p(\mathbf{x}(\omega)/\Omega_j) > p(\mathbf{x}(\omega)/\Omega_k) \Pr\{\Omega_k\}, \qquad (19)$$

$$(\text{all} \quad k, k \neq j).$$

Frequently Eq. (19) is replaced by the stronger condition

$$\Pr\{\Omega_j\} p(\mathbf{x}(\omega)/\Omega_j) > \theta p(\mathbf{x}(\omega)/\Omega_k) \Pr\{\Omega_k\} \qquad (20)$$

$$(\text{all} \quad k, k \neq j),$$

where $\theta > 1$.

By taking logarithms in Eq. (20) we see that the maximum likelihood procedure is equivalent to requiring

$$\sum_{i=1}^{n} x_i \log \frac{p_{ij}(1 - p_{ik})}{p_{ik}(1 - p_{ij})} > \log \theta \left\{ \prod_{i=1}^{n} \frac{(1 - p_{ik})}{(1 - p_{ij})} \right\} \cdot \frac{\Pr\{\Omega_k\}}{\Pr\{\Omega_j\}}. \qquad (21)$$

Setting $a_{ijk} = \log [p_{ij}(1 - p_{ik})/p_{ik}(1 - p_{ij})]$, with $1 \leq j, k \leq m$, $j \neq k$, $1 \leq i \leq n$; and

$$c_{jk} = \log \theta \left\{ \prod_{i=1}^{n} \frac{(1 - p_{ik})}{(1 - pi_j)} \right\} \frac{\Pr \{\Omega_k\}}{\Pr \{\Omega_j\}},$$

with $1 \leq j, k \leq m$, $j \neq k$, we see that (21) requires that $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ lie on the positive side of each of the hyperplanes

$$\sum_{i=1}^{n} a_{ijk} x_i - c_{jk} = 0,$$

and this is class-pair separation. Thus, whenever the maximum likelihood procedure of Equations (19) or (20) results in the correct classification of all points in Ω a class-pair separation will also be successful. The converse is false.

5. Nonlinear separability

Let f be a real-valued continuous function on R^n. We associate with f the surface (in R^n)

$$S_f = \{\mathbf{x} = (x_1, x_2, \cdots, x_n) : f(\mathbf{x}) = 0\}. \tag{22}$$

This surface forms the boundary of the two half-spaces

$$S_f^+ = \{\mathbf{x} : f(\mathbf{x}) > 0\}, \tag{23}$$

and

$$S_f^- = \{\mathbf{x} : f(\mathbf{x}) < 0\}. \tag{24}$$

We shall say that the surface S_f separates the pair (A, B) (of subsets of R^n) provided

$$A \subseteq S_f^+, \qquad B \subseteq S_f^-. \tag{25}$$

Suppose $\mathfrak{A}$ is a family of surfaces in R^n each given by (22) for some f; under what conditions on (A, B) does there exist a surface in $\mathfrak{A}$ which separates (A, B)? For the class of surfaces defined by

$$f(\mathbf{x}) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n - a_0$$

the condition was co $(A) \cap$ co $(B) = \phi$. A natural extension of this class of "linear" surfaces is given by choosing for f a polynomial in $x_1, x_2, \cdots, x_n$, by which is meant an expression of the form

$$f(x_1, x_2, \cdots, x_n) = \sum_{\substack{0 \leq i_j \leq p \\ 1 \leq j \leq n}} a_{i_1 i_2 \cdots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}.$$

The degree of the term $a_{i_1 i_2 \cdots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ is $i_1 + i_2 + \cdots + i_n$. This term is non-zero provided $a_{i_1 i_2 \cdots i_n} \neq 0$ and the degree of f is the maximum of the degrees of its nonzero terms. Let $\mathfrak{A}_k$ denote the class of surfaces given by allowing f to be any polynomial of degree $\leq k$. We have the obvious inclusion relations $\mathfrak{A}_1 \subseteq \mathfrak{A}_2 \subseteq \cdots \subseteq \mathfrak{A}_k \subseteq \mathfrak{A}_{k+1} \subseteq \cdots$ A pair of sets (A, B) is *$\mathfrak{A}_k$-separable* provided there is a polynomial f of degree $\leq k$ such that $A \subseteq S_f^+$ and $B \subseteq S_f^-$.

Two main points regarding separation of classes by polynomial surfaces will now be established.

(1) If A and B are finite sets in R^n with $A \cap B = \phi$, then (A, B) is $\mathfrak{A}_k$-separable for some k.

(2) The algorithm for finding a linear separation can be modified to find an $\mathfrak{A}_k$-separation.

It should be noted that the problem of the design of switching circuits is one in which A and B are subsets of the vertex set of the unit cube $V^n = \{\mathbf{v} = (v_1, v_2, \cdots, v_n) : v_i = 0 \text{ or } 1, 1 \leq i \leq n\}$. The separation of (A, B) by an element of $\{\mathfrak{A}_k\}$ can be viewed as a realization of the switching function π, where

$$\pi(\mathbf{v}) = 1 \quad \text{if } \mathbf{v} \epsilon A, \text{ and } \pi(\mathbf{v}) = 0 \quad \text{if } \mathbf{v} \epsilon B.$$

without determining the explicit logical function involved. The switching functions $\{\pi\}$ which admit an $\mathfrak{A}_1$-separation are just the threshold functions. In Example 3 of Section 5 we illustrate how this procedure may be employed to realize a simple (nonlinearly separable) switching function.

The first point cited above follows from the fact that there exists, by the Lagrange interpolation formula, a polynomial g such that $g = 1$ on A and $g = 0$ on B, provided that $A \cap B = \phi$. In fact, if A and B are subsets of the vertex set of the unit cube, then as is well known, the polynomial has degree $\leq n$.

To prove the assertion of Point 2, as cited above, we consider the mapping $i_{n,k}$ of R^n into $R^{p_{n,k}}$, where

$$p_{n,k} = \sum_{j=1}^{k} \binom{n - 1 + j}{j},$$

which sends the point $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ into the point

$$\left(\cdots \prod_{j=1}^{n} x_j^{i_j} \cdots \right), \text{ where}$$

$$i_j \geq 0,\ 1 \leq j \leq n,\ 1 \leq i_1 + i_2 + \cdots + i_n \leq k.$$

If $k = 2$ then

$$i_{n,2} : \mathbf{x} = (x_1, x_2, \cdots, x_n)$$

$$\rightarrow (x_1^2, x_1 x_2, \cdots, x_1 x_n, x_2^2, x_2 x_3, \cdots,$$

$$x_2 x_n, \cdots, x_{n-1}^2, \cdots, x_{n-1} x_n, x_n^2, x_1, x_2, \cdots, x_n)$$

If we denote by $i_{n,k}(A)$ and $i_{n,k}(B)$ the images of A and B under the mapping $i_{n,k}$, then (A, B) is $\mathfrak{A}_k$-separable if and only if $(i_{n,k}(A), i_{n,k}(B))$ is $\mathfrak{A}_1$-separable. Thus, to find an $\mathfrak{A}_k$-separation of (A, B) we can apply the algorithm of Section 3 to the sets $(i_{n,k}(A), i_{n,k}(B))$.

- *Example 2*

In this example we use $n = 2$, $A = \{\mathbf{a}_1 = (0, 0), \mathbf{a}_2 = (1, 1)\}$,

$A = \{\mathbf{a}_1 = (0, 0), \mathbf{a}_2 = (1, 1)\}$, and

$B = \{\mathbf{b}_1 = (0, 1), \mathbf{b}_2 = (1, 0)\}$.

Both co (A) and co (B) contain the point $(1/2, 1/2)$ and hence (A, B) is not $\mathfrak{A}_1$-separable. We now seek an $\mathfrak{A}_2$-separation. Since the sets A and B are subsets of the vertex set of the unit cube V^2 it suffices to employ the mapping $\wedge : \mathbf{x} = (x_1, x_2) \rightarrow (x_1, x_1, x_2, x_2)$.
The sets A and B are mapped into the sets

$\hat{A} = \{\hat{\mathbf{a}}_1 = (0, 0, 0), \hat{\mathbf{a}}_2 = (1, 1, 1)\}$,

$\hat{B} = \{\hat{\mathbf{b}}_1 = (0, 0, 1), \hat{\mathbf{b}}_2 = (1, 0, 0)\}$.

To each of the vectors in $\hat{A} \cup \hat{B}$ we adjoin a fourth coordinate equal to 1, obtaining finally

$A^* = \{\mathbf{a}_1^* = (0, 0, 0, 1), \mathbf{a}_2^* = (1, 1, 1, 1)\}$,

$B^* = \{\mathbf{b}_1^* = (0, 0, 1, 1), \mathbf{b}_2^* = (1, 0, 0, 1)\}$.

We take $\theta = 1/2$ and employ the training set $\mathfrak{T} = \{\mathbf{a}_1^*, \mathbf{a}_2^*, \mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{a}_1^*, \mathbf{a}_2^*, \mathbf{b}_1^*, \mathbf{b}_2^*, \cdots\}$,

obtaining the sequence below.

$\mathbf{w}_0 = (0, 0, 0, 0)$

$\mathbf{w}_1 = (0, 0, 0, 1) = \mathbf{w}_0 + \mathbf{a}_1^*$

$\mathbf{w}_2 = (0, 0, 0, 1) = \mathbf{w}_1$

$\mathbf{w}_3 = (0, 0, -1, 0) = \mathbf{w}_2 - \mathbf{b}_1^*$

$\mathbf{w}_4 = (-1, 0, -1, -1) = \mathbf{w}_3 - \mathbf{b}_2^*$

$\mathbf{w}_5 = (-1, 0, -1, 0) = \mathbf{w}_4 + \mathbf{a}_1^*$

$\mathbf{w}_6 = (0, 1, 0, 1) = \mathbf{w}_5 + \mathbf{a}_2^*$

$\mathbf{w}_7 = (0, 1, -1, 0) = \mathbf{w}_6 - \mathbf{b}_1^*$

$\mathbf{w}_8 = (-1, 1, -1, -1) = \mathbf{w}_7 - \mathbf{b}_2^*$

$\mathbf{w}_9 = (-1, 1, -1, 0) = \mathbf{w}_8 + \mathbf{a}_1^*$

$\mathbf{w}_{10} = (0, 2, 0, 1) = \mathbf{w}_9 + \mathbf{a}_2^*$

$\mathbf{w}_{11} = (0, 2, -1, 0) = \mathbf{w}_{10} - \mathbf{b}_1^*$

$\mathbf{w}_{12} = (-1, 2, -1, -1) = \mathbf{w}_{11} - \mathbf{b}_2^*$

$\mathbf{w}_{13} = (-1, 2, -1, 0) = \mathbf{w}_{12} + \mathbf{a}_1^*$

$\mathbf{w}_{14} = (0, 3, 0, 1) = \mathbf{w}_{13} + \mathbf{a}_2^*$

$\mathbf{w}_{15} = (0, 3, -1, 0) = \mathbf{w}_{14} - \mathbf{b}_1^*$

$\mathbf{w}_{16} = (-1, 3, -1, -1) = \mathbf{w}_{15} - \mathbf{b}_2^*$

$\mathbf{w}_{17} = (-1, 3, -1, 0) = \mathbf{w}_{16} + \mathbf{a}_1^*$

$\mathbf{w}_{18} = (-1, 3, -1, 0) = \mathbf{w}_{17}$

$\mathbf{w}_{19} = (-1, 3, -1, 0) = \mathbf{w}_{18}$

$\mathbf{w}_{20} = (-1, 3, -1, 0) = \mathbf{w}_{19}$

$\mathbf{w}_{21} = (-1, 3, -1, 1) = \mathbf{w}_{20} + \mathbf{a}_1^*$

$\mathbf{w}_{22} = (-1, 3, -1, 1) = \mathbf{w}_{21}$

$\mathbf{w}_{23} = (-1, 3, -2, 0) = \mathbf{w}_{22} - \mathbf{b}_1^*$

$\mathbf{w}_{24} = (-1, 3, -2, 0) = \mathbf{w}_{23}$

$\mathbf{w}_{25} = (-1, 3, -2, 1) = \mathbf{w}_{24} + \mathbf{a}_1^*$

$\mathbf{w}_{26} = (-1, 3, -2, 1) = \mathbf{w}_{25}$

$\mathbf{w}_{27} = (-1, 3, -2, 1) = \mathbf{w}_{26}$

$\mathbf{w}_{28} = (-2, 3, -2, 0) = \mathbf{w}_{27} - \mathbf{b}_2^*$

$\mathbf{w}_{29} = (-2, 3, -2, 1) = \mathbf{w}_{28} + \mathbf{a}_1^*$

$\mathbf{w}_{30} = (-1, 4, -1, 2) = \mathbf{w}_{29} + \mathbf{a}_2^*$

$\mathbf{w}_{31} = (-1, 4, -2, 1) = \mathbf{w}_{30} - \mathbf{b}_1^*$

$\mathbf{w}_{32} = (-2, 4, -2, 0) = \mathbf{w}_{31} - \mathbf{b}_2^*$

$\mathbf{w}_{33} = (-2, 4, -2, 1) = \mathbf{w}_{32} + \mathbf{a}_1^*$

$\mathbf{w}_{34} = (-2, 4, -2, 1) = \mathbf{w}_{33}$

$\mathbf{w}_{35} = (-2, 4, -2, 1) = \mathbf{w}_{34}$

$\mathbf{w}_{36} = (-2, 4, -2, 1) = \mathbf{w}_{35}$

$\mathbf{w}_{36} = \mathbf{w}_{37} = \cdots$

The hyperbola $(2x_1 - 1)(2x_2 - 1) - c = 0$ separates (A, B) for every c satisfying $-1/2 < c < 1/2$.

• *Example 3*

This example pertains to the most significant digit in the product of two integers. Let $V^n = \{\mathbf{v} = (v_1, v_2, \cdots, v_n): v_i = 0 \text{ or } 1, 1 \leq i \leq n\}$. With each $\mathbf{v} \epsilon V^n$ we associate the integer $N_n(\mathbf{v})$, where

$$N_n(\mathbf{v}) = v_1 + 2v_2 + 4v_3 + \cdots + 2^{n-1}v_n.$$

Consider the mapping $*$ from $V^n \times V^n$ into V^{2n} defined by

$$\mathbf{x} = (x_1, x_2, \cdots, x_n), \qquad \mathbf{y} = (y_1, y_2, \cdots, y_n)$$
$$z = (z_1, z_2, \cdots, z_{2n}).$$

Then $\mathbf{x} * \mathbf{y} = \mathbf{z}$ if $N_n(\mathbf{x}) \times N_n(\mathbf{y}) = N_{2n}(\mathbf{z})$. Finally we consider the truth function f (defined on $V^n \times V^n$) which is equal to the most significant digit in the binary expansion of $N_n(\mathbf{x}) \times N_n(\mathbf{y})$, that is,

$$f(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 \text{ if } z_{2n} = 1 \\ 0 \text{ if } z_{2n} = 0. \end{cases}$$

Let $\Sigma_+ = \{(\mathbf{x}, \mathbf{y}): f(\mathbf{x}, \mathbf{y}) = 1\}$ and $\Sigma_- = \{(\mathbf{x}, \mathbf{y})_0 f(\mathbf{x}, \mathbf{y}) = 0\}$. The sets Σ_+ and Σ_- are quadratically separable.

This is seen by noting that

$$f(x, y) = \begin{cases} 1 \text{ if } (x_1 + 2x_2 + \cdots + 2^{n-1}x_n) \\ \quad \times (y_1 + 2y_2 + \cdots + 2^{n-1}y_n) \geq 2^{2n-1} \\ 0 \text{ if otherwise.} \end{cases}$$

Thus the quadratic surface

$$S_\epsilon = \left\{(\mathbf{x}, \mathbf{y}) : \sum_{i,j=1}^{n} x_i y_j 2^{i+j-2} - (2^{2n-1} + \epsilon) = 0\right\}$$

separates (Σ_+, Σ_-) for every ϵ, $0 < \epsilon < 1$. Employing the linear separation algorithm to find such a quadratic surface for $n = 2$ we obtain the surface

$$-x_1 + 4x_1x_2 + 3x_1y_1 + 3x_1y_2 - x_2 + 3x_2y_1 + 3x_2y_2 - 2y_1 + 2y_1y_2 - 3y_2 - c = 0,$$

which separates (Σ_+, Σ_-) for every c with $8 < c < 10$.

6. Experimental results

In any attempt to classify patterns represented by a set of numerical measurements through the use of linear separability and one of the algorithms presented, three questions take on basic importance. These are stated and discussed briefly below.

First, are the classes linearly separable? This question is answered by applying the algorithm for determining the separating hyperplanes, with the knowledge that if separation is possible, the hyperplanes will be found; that is, the algorithm will terminate.

Second, how long will it take to determine the separating hyperplane? It is not practical to estimate this time requirement in advance; only experiment can yield this information. Clearly the time is a function of the number of patterns used in the training process. Usually one finds it practical to devote considerable computer time to finding the separations, for this is done in advance of use of the computer for pattern recognition and need not be repeated for the same space of patterns.

Third, how well will the separating hyperplane separate patterns which are not included in the training set but which are to be separated into the same classes? This is a question involving the error rate on new patterns and one that can be answered only by subsequent testing of the new data. On the other hand, since the linear functionals are continuous it is reasonable to assume that new patterns that are close to old patterns will be similarly classified. The limits of such a form of 'generalization' depend upon how well the linear functionals separate the classes. There is a well-defined and obvious sense by which we may rank the functionals which separate the two classes A and B. However, the algorithm given in Section 3 yields only one such separating functional and the problem of finding the "best" separation is a difficult unsolved problem.

A number of experiments have been performed in which the patterns to be separated were sets of typed or printed alphanumeric characters. In some tests the measurements consisted of raw data taken from the output of a cathode ray tube scanner which presented the data as a matrix of zeros and ones. In other tests this data was preprocessed and each character was finally represented by a binary vector, each component of which designated the presence or absence of a certain "feature" in that character. Except as noted, the classes were taken to be the letters and numbers themselves, i.e. upper case A's constituted one class, lower case a's another, etc. In a single class might be included several styles of type for the same letter or number, and a number of samples of each style on which to train.

Some discussion of results from typical experiments follows. Although the accuracy on new data will be seen to be generally good, this is not of primary interest here; rather the examples are given to illustrate the different kinds of character recognition tasks that can be accomplished by linear separation. All of the tests made use of Procedure 1, as described in Section 4, and the input was from data stored on tapes furnished by the Engineering Sciences Department of the IBM Research Division.

• *Some typical experiments*

1. In one experiment the raw data input was in the form of a 32×32 binary scanner output representing alphabetic characters from IBM EXECUTIVE® typewriters with a variety of typefaces. Characters were scanned as they appeared in original documents and in Bruning, Verifax, Xerox, and carbon copies. Training involved 10 upper case alphabets, testing involved 9. In a typical result for $\Theta = 100$, the required 26 linear functions were obtained in 9.2 minutes of IBM 7090 time, with an error rate of 1.7% for all inputs and an error rate of 0.7% for inputs derived from typed (unreproduced) documents.

2. A related experiment involved original data, as above, but generated binary search separations. In this experiment a different classification of the alphabet was attempted by linear separation. Here linear functionals were formed to successively split the alphabet into two groups, each of those two more, and so on until individual letters were identified. Linear separability was demonstrated. This classification scheme has the advantage of cutting the generation time of the functionals by a factor of $(\log_2 M)/(M)$ if M functionals are required. If the functionals are tested sequentially to determine the classification of a pattern, as is the case on a computer, then this scheme requires only $\log_2 M$ tests in place of $M - 1$. The error rate was comparable to that for the ordinary letter-by-letter testing mentioned above. For $\Theta = 100$ the generation time on

® A trademark of the International Business Machines Corporation.

the above example for the binary search separations was 2.07 minutes.

3. The object of an experiment with a raw data input was to separate upper from lower case characters. It involved 5 upper case and 5 lower case IBM EXECUTIVE typewriter alphabets and 10 samples of copies used for training. Reading on 20 new alphabets, an error rate of 0.4% was obtained using only a single linear functional to separate the two classes (upper and lower cases) for all the alphabets involved. The generation time for this functional was 0.94 minutes.

4. This experiment involved a raw data input also and was concerned with the separation of upper and lower case characters in a variety of IBM SELECTRIC® typewriter fonts. Here a single linear functional was generated to distinguish upper case from lower case letters in 5 fonts (4 of which contained both upper and lower case letters). The fonts included SCRIBE, a script font. Since only training data was available, no error rate for new data was established.

5. This experiment was concerned with the binary features of data from a Russian journal. The input comprised 108 binary components and training employed some 30 samples of each of 32 letters. The generation time for 32 functionals was 5.03 minutes. The error rate in reading 4013 new characters was 0.3% with some sections of tape exhibiting a considerably lower error rate (1 error in the first 800 characters, 4 in the next 1800, and 7 in the last 1413).

6. Here we were concerned with binary features of multifont data. Each character was represented by a feature (vector) of 96 binary measurements. Twenty sets of characters were taken for training. Each set consisted of: (a) 26 upper case alphabetic characters, (b) 26 lower case alphabetic characters, and (c) 9 numeric characters, with all 61 characters appearing in three IBM SELECTRIC typewriter fonts (ELITE, ADJUTANT, and SCRIBE). Thus, a total of $3 \times 61 \times 20 = 3660$ characters were used in the 'training' phase. Sixty-one linear functionals were constructed during the training phase, which required 30 minutes on an IBM 7094. These functionals were in turn tested on a new group of 20 sets of alpha-numeric characters with the same composition as the training set. There resulted 3 errors for an error rate of 0.0823%.

7. Finally, we summarize an experiment with video multifont data. Each character was represented by a 17×32 matrix with elements zero and one. A total of 20 sets, each consisting of twenty-six upper case alphabetic characters in each of three IBM SELECTRIC typewriter fonts (PICA, ADVOCATE, and DELEGATE) were used in training. Thus a total of $20 \times 26 \times 3 = 1560$ characters were used in training. Twenty-six linear functionals were determined in 25 minutes of IBM 7094 computation. These linear functionals were then tested on 39 sets of similar composition, resulting in 19 errors for an error rate of 0.0625%.

Acknowledgments

The authors express their appreciation to Mrs. Carol Wade and Mr. Otto Mond for their help in programming the experimental results reported in this paper. Thanks are also due to Dr. L. P. Horwitz for helpful discussions on character recognition problems and on the choice of experiments.

References

1. H. D. Block, B. W. Knight, Jr., and F. Rosenblatt, "Analysis of a Four Layer Series-Coupled Perceptron," *Reviews of Modern Physics* **34,** 135 (1962).
2. F. Rosenblatt, *Principles of Neurodynamics*, Cornell Aeronautical Laboratory, 1961.
3. G. Palmieri and R. Sanna, *Automatic Probabilistic Programmer/Analyzer for Pattern Recognition*, Istituto di Fisica, University of Genoa, unpublished.
4. B. Widrow and J. Angell, "Reliable, Trainable Networks for Computing and Control," *Aerospace Engineering* **21,** 78 (September, 1962).
5. J. S. Griffin, Jr., J. H. King, Jr., and C. J. Tunis, "A Pattern Identification System Using Linear Decision Functions," *IBM Systems Journal* **2,** 548 (September–December, 1963).
6. A. Novikoff, *On Convergence Proofs for Perceptrons*, Stanford Research Institute, January, 1963.

Received January 10, 1964