Generation of Polynomial Discriminant Functions for Pattern Recognition

DONALD F. SPECHT, MEMBER, IEEE

Abstract—A practical method of determining weights for cross-product and power terms in the variable inputs to an adaptive threshold element used for statistical pattern classification is derived. The objective is to make it possible to realize general nonlinear decision surfaces, in contrast with the linear (hyperplanar) decision surfaces that can be realized by a threshold element using only first-order terms as inputs. The method is based on nonparametric estimation of a probability density function for each category to be classified so that the Bayes decision rule can be used for classification. The decision surfaces thus obtained have good extrapolating ability (from training patterns to test patterns) even when the number of training patterns is quite small. Implementation of the method, both in the form of computer programs and in the form of polynomial threshold devices, is discussed, and some experimental results are described.

Index Terms—Bayes strategy, density functions, discriminant functions, estimation of probability, implementation, machine learning, nonlinear, nonparametric, polynomial, statistical pattern classification.

Introduction

PATTERN classification techniques are useful in many practical problems such as weather forecasting, medical diagnosis, adaptive control, and speech recognition [2], [3]. A considerable body of literature exists concerning the use of adaptive linear and piecewise linear threshold elements for pattern classification [2]–[11].

The purpose of any attempt at pattern recognition is the identification of the underlying characteristics which are common to a class of objects. Correct identification of the underlying characteristics enables one to extrapolate; i.e., to identify a new object as belonging to a certain class on the basis of its underlying characteristics and in spite of variations of incidental characteristics within the class. It is often the case, however, that the variations of incidental characteristics of a class of objects are of such a magnitude as to obscure the underlying characteristics. In such cases it is frequently advisable to use some type of statistical technique to discover the underlying characteristics. These characteristics can be specified in detail in terms of probability density functions (assuming that the patterns to be classified are drawn from some statistical population).

It is easily shown that the optimal (in the Bayes sense) decision surface for separation of patterns in two categories whose probability density functions[1] have normal (Gaussian) distributions with equal covariance matrices is a hyperplane (see for example Koford and Groner [8]); but for most practical problems the Bayes optimal decision surfaces, if known, would not be hyperplanes. It is therefore desirable to be able to determine nonlinear decision surfaces which can closely approximate the Bayes decision surface regardless of its shape. Procedures for doing this have been discussed by Cover and Hart (the nearest-neighbor decision rule) [12] and by Sebestyen (adaptive sample-set construction) [13]. Although these procedures are quite useful for certain classes of problems, the former requires storage of all training patterns and a relatively large amount of calculation to classify new patterns, and the latter requires estimation of an arbitrary probability density function as the sum of a relatvely small number of Gaussian density functions, and involves *ad hoc* rules for the grouping of training samples. These features make analysis quite difficult and can be responsible for significant errors in some problems.

The *polynomial discriminant method* (PDM) proposed in this paper is a practical method for determining nonlinear decision surfaces, which is based on the Bayes decision rule, but which avoids the above difficulties. No storage of training patterns is required since training is accomplished on a one-pattern-at-a-time basis and does not require iteration. Classification of new patterns requires evaluation of a single polynomial for each possible category, or evaluation of one polynomial (which represents the decision boundary) for a two-category problem.

The Bayes Strategy

An accepted norm for decision rules or strategies which are used to classify patterns is that they do so in such a way as to minimize the "expected risk." Such strategies are called "Bayes strategies" [14] and may be applied to problems containing any number of categories.

Consider the two-category situation in which the state of nature θ is known to be either θ_A or θ_B. If it is desired to decide whether $\theta=\theta_A$ or $\theta=\theta_B$ based on a set of measurements represented by the p-dimensional vector $X^t=[X_1 \cdots X_j \cdots X_p]$, it can be shown (Sebes-

Manuscript received August 5, 1966; revised January 16, 1967. This paper is based on the author's Ph.D. dissertation [1], Stanford University, Stanford, Calif. The research was sponsored by the Lockheed Missiles & Space Company.

The author is with the Lockheed Palo Alto Research Lab., Palo Alto, Calif.

[1] Often called "densities" for convenience.

tyen [15], p. 32) that the Bayes strategy leads to the (Bayes) decision rule

$$\left.\begin{aligned} d(X) &= \theta_A \quad \text{if } h_A l_A f_A(X) > h_B l_B f_B(X) \\ d(X) &= \theta_B \quad \text{if } h_A l_A f_A(X) < h_B l_B f_B(X) \end{aligned}\right\} \tag{1}$$

where $f_A(X)$ and $f_B(X)$ are the probability density functions for categories θ_A and θ_B respectively, l_A is the loss associated with the decision $d(X) = \theta_B$ when $\theta = \theta_A$, l_B is the loss associated with the decision $d(X) = \theta_A$ when $\theta = \theta_B$ (the losses associated with correct decisions are taken to be equal to zero), h_A is the a priori probability of occurrence of patterns from category θ_A, and $h_B = 1 - h_A$ is the a priori probability that $\theta = \theta_B$. Thus the boundary between the region in which the Bayes decision $d(X) = \theta_A$ and the region in which $d(X) = \theta_B$ is given by the equation

$$f_A(X) = K f_B(X) \tag{2}$$

where

$$K \triangleq \frac{h_B l_B}{h_A l_A} \cdot \tag{3}$$

Note that in general the two-category decision surface defined by (2) can be arbitrarily complex, since there is no restriction on the densities $f_A(X)$ and $f_B(X)$ except those conditions which all probability density functions must satisfy; namely, that they are everywhere non-negative, that they are integrable, and that their integrals over all space equal unity.

Now consider the q-category problem in which $d(X) = \theta_r$ where $1 \leq r \leq q$. Again, let the losses associated with correct decisions be zero and let the losses associated with incorrect decisions be positive. The values of these latter losses are dictated by the consequences of incorrect decisions in the classification problem to be solved and must be assigned as part of the problem definition. For simplicity of results, apply the restriction that losses associated with all incorrect decisions given $\theta = \theta_r$ are equal, and let those losses be denoted by l_r. Then the Bayes decision rule becomes $d(X) = \theta_r$ such that

$$h_r l_r f_r(X) \geq h_s l_s f_s(X) \quad \text{for all } s \neq r. \tag{4}$$

In order to use the Bayes decision rules to define a decision boundary in pattern recognition, it is necessary to know or estimate the a priori probabilities h_r, the losses l_r and the densities $f_r(X)$. Often the a priori probabilities are known or can be estimated accurately. The requirements for estimation of the loss function l allows us to make either a subjective evaluation or, in some problems, a numerical calculation of the relative importance of misclassification errors given each of the various states of nature. However, it is a rare practical problem in which the densities are known.

Estimation of a Probability Density Function as a Sum of Exponentials

Since it has been established that use of a Bayes strategy for pattern classification involves estimating the probability density functions of each of the categories, the analysis will begin by considering means of estimating these densities for each category, first as a sum of exponentials, and then as a single polynomial.

If the probability densities of the patterns in the categories to be separated are unknown, and all that is given is a set of training patterns (training samples), then it is these samples which provide the only clue to the unknown underlying probability densities ("parent" densities). If the number of training samples is limited, with no possibility of obtaining more, it is not known that there is a one best way of utilizing these samples to estimate the density functions from which they came. Thus it is necessary to make assumptions concerning the form of the density functions. All that is really known about the parent density is that the given samples have a positive probability of occurring. If the density function is continuous, it is usually reasonable to assume that points not in the training set, but near a given sample point, have about the same probability of occurring as does the given point. This assumption is equivalent to an assumption that the density function is smooth and that the first partial derivatives of the density function are small. (This assumption will henceforth be referred to as Assumption 1.)

An interpolation function, $g(X, X_i)$, will be found such that the overall parent density can be estimated adequately by

$$f(X) = \frac{1}{m} \sum_{i=1}^{m} g(X, X_i), \tag{5}$$

where m is the number of training patterns available and $g(X, X_i)$ is the contribution of the ith training pattern to the estimated density. The purpose of the interpolation function is to assure that Assumption 1 is valid for the estimated density since it is presumed to be valid for the (parent) density being estimated.

The following are desirable characteristics for the interpolation function.

1) The contribution of one pattern to the overall density should not be dependent on other patterns in the training set. This characteristic is essential to achieve the practical goal of being able to utilize patterns one at a time, rather than requiring storage of the entire training set before training can be accomplished.

2) The overall estimated density function should be expressible analytically and should be smooth and continuous.

3) The interpolation function must approach zero as distance from the training sample approaches infinity.

Condition 1 implies that the interpolation function should be a function of the Euclidean distance of the point X from the training point i (i.e., the interpolation

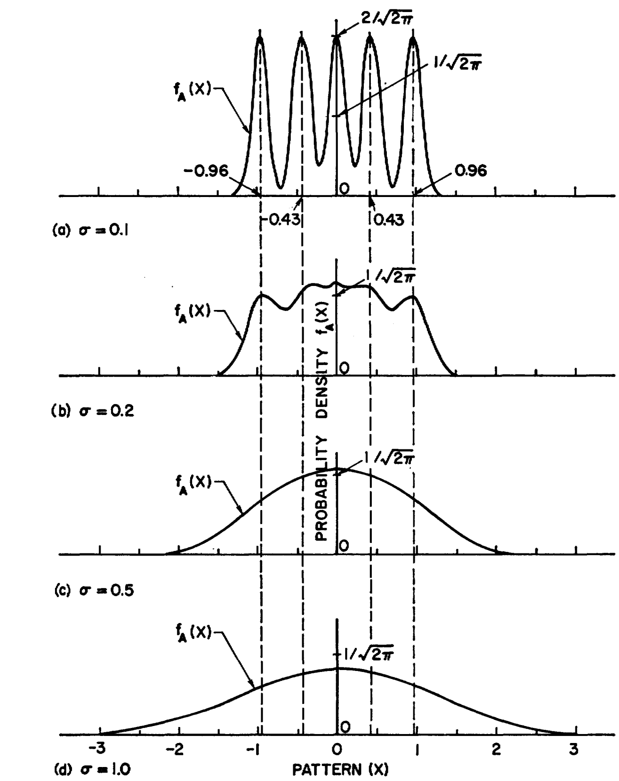

Fig. 1. The smoothing effect of σ on estimated parent probability density function, $f_A(X)$ for five training samples at $X = +0.96$, $+0.43$, 0.00, -0.43, and -0.96. Estimate of $f_A(X)$ is from (7).

function should have spherical symmetry, since given only a single training point, the only parameter which relates another point to this training point is its distance therefrom). Thus $g(X, X_i)$ is restricted to being a function of $X - X_i$ and can be written as $g(X - X_i)$. From among the many functions which satisfy conditions 1, 2, and 3, the interpolation function

$$g(X - X_i) \triangleq \frac{1}{(2\pi)^{p/2}\sigma^p} \cdot \exp\left[-\frac{(X - X_i)^t(X - X_i)}{2\sigma^2} \right], \quad (6)$$

where σ is a "smoothing parameter," was chosen for use in this study because it has a number of desirable properties, the most important of which is that it leads to a practical method for calculating the coefficients of a polynomial discriminant function.

Let the estimated probability density function $f(X)$ for some category θ_A be denoted by $f_A(X)$. Then, using the interpolation function given by (6), $f_A(X)$ becomes the estimator[2]

$$f_A(X) = \frac{1}{\sigma^p(2\pi)^{p/2}} \frac{1}{m} \cdot \sum_{i=1}^{m} \exp\left[-\frac{(X_{ai} - X)^t(X_{ai} - X)}{2\sigma^2} \right], \quad (7)$$

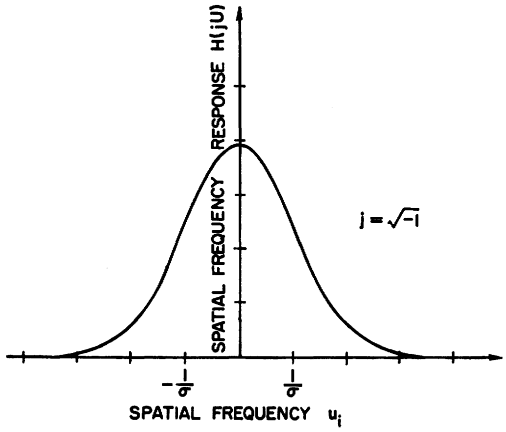

Fig. 2. Spatial frequency response $H(jU)$.

where

i = pattern number, and

X_{ai} = ith training pattern from category θ_A,

$$\triangleq \begin{bmatrix} X_{ai1} \\ \cdot \\ \cdot \\ X_{aij} \\ \cdot \\ \cdot \\ X_{aip} \end{bmatrix}.$$

Figure 1 illustrates the effect of σ on $f_A(X)$ in the case in which the independent variable X is one-dimensional. The density $f_A(X)$ is plotted from (7) for four values of σ, with the same 5 training samples in each case. A value of $\sigma = 0.1$ (Fig. 1(a)) causes the estimated parent density function to have 5 distinct modes corresponding to the 5 training samples. $\sigma = 0.2$ brings about a greater degree of interpolation between points, but the modes are still distinct. With $\sigma = 0.5$, $f_A(X)$ has but a single mode and a shape approximating that of the Gaussian distribution, which could very well be the distribution of the parent density function from which these samples were taken. The value of $\sigma = 1.0$ causes some flattening of the density function, with spreading out of the tails.

The smoothing effect of σ evident in Fig. 1 can be better understood by the following analysis:[3] The probability density of the samples from category θ_A may be estimated as a Dirac delta function of volume $1/m$ at each of the sample points and as zero elsewhere. Let this estimate be denoted by $f_A'(X)$. Then the estimate of $f_A(X)$ given by (7) is equivalent to the convolution of $f_A'(X)$ with $h(X)$ where

$$h(X) \triangleq \frac{1}{\sigma^p(2\pi)^{p/2}} \exp\left(-\frac{X^tX}{2\sigma^2} \right). \quad (8)$$

Then $h(X)$ can be interpreted as the impulse response of a filter acting on the probability density $f_A'(X)$ of the samples. The spatial frequency response of the filter having an impulse response $h(X)$ is given by the Fourier integral of $h(X)$

[2] This is the estimator proposed by Sebestyen [13] in the case for which each training sample is the mean of a separate subclass (i.e., before grouping samples into a smaller number of subclasses).

[3] Suggested by Karl Belser, Dept. Elec. Engrg., Stanford University, Stanford, Calif.

$$
\begin{aligned}
H(jU) &\overset{\Delta}{=} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \exp\,(-ju_1X_1 - \cdots \\
&\quad -ju_pX_p)h(X)dX_1 \cdots dX_p \\
&= (2\pi)^{p/2} \exp\,(-\tfrac{1}{2}\sigma^2U^tU),
\end{aligned} \tag{9}
$$

where $U^t = [u_1 \cdots u_i \cdots u_p]$, and $j = \sqrt{-1}$. Since $H(jU)$ is spherically symmetric about the origin, any cross section through the origin has the shape given in Fig. 2.

From the foregoing discussion it is seen that the use of the interpolation function $g(X - X_i)$ to estimate $f_A(X)$ has the effect of a spatial low-pass filter and that the cutoff frequency of the filter is inversely proportional to σ.

Equation (7) can be used directly with the decision rules expressed by (1) (two-category problem) or by (4) (many-category problem). A computer program has been written for the more general case, and excellent results have been obtained using this method on practical pattern-recognition problems. One of these problems is discussed in detail in Specht [16]. However, two practical limitations are inherent in the use of (7). First, the entire training set must be stored (either in a computer memory or in special-purpose hardware) and used during testing; and second, the amount of computation necessary to classify an unknown point is proportional to the size of the training set. Neither of these limitations is important if the size of the training set is small. However, the frequent requirement for a large training set leads one to look for more efficient classifiers having the same general applicability as those expressed by (4) and (7). A classifier with these characteristics will be defined in the next section.

The Polynomial Discriminant Function

To implement a separating boundary which may be nonlinear, consider the use of the polynomial discriminant function

$$
\begin{aligned}
P(X) = D_{0\cdots 0} &+ D_{10\cdots 0}X_1 + D_{010\cdots 0}X_2 + \cdots + D_{0\cdots 01}X_p \\
&+ D_{20\cdots 0}X_1^2 + D_{110\cdots 0}X_1X_2 + \cdots \\
&+ D_{z_1z_2\cdots z_p}X_1^{z1}X_2^{z2} \cdots X_p^{zp} + \cdots .
\end{aligned} \tag{10}
$$

This function, which mechanizes a completely general nonlinear decision surface if $P(X) = 0$ is taken as the boundary between classification of X as belonging to category θ_A versus category θ_B, has a strong appeal of simplicity for testing new points so long as the number of terms required in (10) can be held to a practical limit.

From (7), the density for category θ_A is

$$
\begin{aligned}
f_A(X) &= \frac{1}{\sigma^p(2\pi)^{p/2}} \frac{1}{m} \sum_{i=1}^{m} \exp\left[-\frac{(X_1 - X_{ai1})^2 + \cdots + (X_j - X_{aij})^2 + \cdots + (X_p - X_{aip})^2}{2\sigma^2}\right] \\
&= \frac{1}{\sigma^p(2\pi)^{p/2}} \frac{1}{m} \sum_{i=1}^{m} \exp\left(-\frac{X_1^2 + \cdots + X_j^2 + \cdots + X_p^2}{2\sigma^2}\right) \\
&\quad \cdot \exp\left(-\frac{-2X_1X_{ai1} + X_{ai1}^2 - \cdots - 2X_jX_{aij}^2 + X_{aij} - \cdots - 2X_pX_{aip} + X_{aip}^2}{2\sigma^2}\right).
\end{aligned} \tag{11}
$$

Letting

$$
B_{ai} \overset{\Delta}{=} -\frac{1}{2} \sum_{j=1}^{p} X^2{}_{aij},
$$

(11) becomes

$$
f_A(X) = \frac{1}{\sigma^p(2\pi)^{p/2}} \left[\exp\left(-\frac{X^tX}{2\sigma^2}\right)\right] \frac{1}{m} \sum_{i=1}^{m} \exp\left[\frac{(X_1X_{ai1} + \cdots + X_jX_{aij} + \cdots + X_pX_{aip}) + B_{ai}}{\sigma^2}\right]. \tag{12}
$$

Using the Taylor's series expansion and the multinomial theorem there results:

$$
\begin{aligned}
&\frac{1}{m} \sum_{i=1}^{m} \exp\left[\frac{X_1X_{ai1} + \cdots + X_pX_{aip} + B_{ai}}{\sigma^2}\right] \\
&= \frac{1}{m} \sum_{i=1}^{m} \left[\exp\left(\frac{B_{ai}}{\sigma^2}\right)\right] \cdot \left[1 + \frac{X_1X_{ai1} + \cdots + X_pX_{aip}}{\sigma^2} + \frac{X_1^2X_{ai1}^2 + 2X_1X_2X_{ai1}X_{ai2} + 2X_1X_3X_{ai1}X_{ai3} + \cdots}{2!\sigma^4}\right. \\
&\quad + \frac{X_1^3X_{ai1}^3 + 3X_1^2X_2X_{ai1}^2X_{ai2} + 6X_1X_2X_3X_{ai1}X_{ai2}X_{ai3} + \cdots}{3!\sigma^6} + \cdots \\
&\quad \left. + \frac{\dfrac{h!}{z_1!z_2! \cdots z_p!} X_1^{z_1} X_2^{z_2} \cdots X_p^{z_p} X_{ai1}^{z_1} X_{ai2}^{z_2} \cdots X_{aip}^{z_p}}{h!\sigma^{2h}} + \cdots \right],
\end{aligned} \tag{13}
$$

where z_j is the non-negative integer exponent of X_j and $z_1 + \cdots + z_j + \cdots + z_p = h$.

Equation (12) can be written in the form

$$f_A(X) = \frac{1}{\sigma^p(2\pi)^{p/2}}\left[\exp\left(-\frac{X^tX}{2\sigma^2}\right)\right]P^A(X), \quad (14)$$

where

$$\begin{aligned}P^A(X) = {} & D^A_{0\cdots 0} + D^A_{10\cdots 0}X_1 + D^A_{010\cdots 0}X_2 + \cdots \\ & + D^A_{0\cdots 01}X_p + D^A_{20\cdots 0}X_1^2 + D^A_{110\cdots 0}X_1X_2 \\ & + \cdots + D^A_{z_1z_2\cdots z_p}X_1^{z_1}X_2^{z_2}\cdots X_p^{z_p} + \cdots \quad (15)\end{aligned}$$

Note that $P^A(X)$ is in the form of the proposed polynomial discriminant function given by (10). Since (15) involves patterns from only one category θ_A, it is apparent that a polynomial of the form given in (15) will be needed for each category in a many-category discrimination problem.

The coefficients $D^A_{z_1z_2\cdots z_p}$ in (15) are the same as the coefficients of the cross products $X_1^{z_1}X_2^{z_2}\cdots X_p^{z_p}$ in (13). Thus,

$$\begin{aligned}D^A_{z_1z_2\cdots z_p} = {} & \frac{1}{z_1!z_2!\cdots z_p!\sigma^{2h}}\,\frac{1}{m} \\ & \cdot \sum_{i=1}^{m} X^{z_1}_{ai1}X^{z_2}_{ai2}\cdots X^{z_p}_{aip}\exp\left(\frac{B_{ai}}{\sigma^2}\right) \quad (16)\end{aligned}$$

where $h = z_1 + z_2 + \cdots + z_p$, and $B_{ai} \triangleq -\frac{1}{2}X^t_{ai}X_{ai}$.

Equation (16) provides a method for calculating the coefficients of the desired polynomial discriminant function, (15), based on training patterns. The implementation of these equations constitute an algorithm which may be called a "training algorithm." Although the generality of the notation used makes the equations look formidable, consider the coefficient of a specific term in (15) such as $D^A_{110}\cdots{}_0X_1X_2$. From (16),

$$D^A_{110\cdots 0} = \frac{1}{\sigma^4}\sum_{i=1}^{m} X_{ai1}X_{ai2}\exp\left(\frac{B_{ai}}{\sigma^2}\right).$$

In words, this equation says to take the average of the products of the cross products $X_{ai1}X_{ai2}$ and a "normalizing factor" $\exp(B_{ai}/\sigma^2)$; then to multiply this average by a "premultiplying constant" $1/\sigma^4$. Each term has its own premultiplying constant. Note that all terms for a pattern X_{ai} have the same normalizing factor. The normalizing factor, therefore, need be calculated only once for each training pattern, regardless of the number of coefficients used in the polynomial $P^A(X)$. Considering this circumstance, and also the fact that the premultiplying constant is not data-dependent, the training algorithm implied by (16) is very similar to that of the adaptive matched filter in which each coefficient in the equation

$$M(X) = W_1X_1 + \cdots + W_pX_p \quad (17)$$

is made equal to the mean of the corresponding variable for the patterns in the training set.

Use of Polynomial Discriminant Functions for Pattern Recognition

Relating the results just obtained to the many-category discrimination problem, the optimum decision rule, (4), can now be rewritten, using (14), as follows.

$$\text{Choose } d(X) = \theta_r \text{ such that } h_rl_rP^r(X) \geqq h_sl_sP^s(X) \quad \text{for all } s \neq r, \quad (18)$$

where $P^r(X)$ and $P^s(X)$ are given by (15) for categories θ_r and θ_s, respectively.

Classification of points X into one of many categories θ_r can be accomplished by means of the decision rule (18), where $P^r(X)$ and $P^s(X)$ are defined by (15) with appropriate changes in superscripts, and the coefficients of (15) are calculated using (16) (again with appropriate changes in superscripts, and also in the subscript a). This method of classification of points X will be designated the "polynomial discriminant method" (PDM) of pattern recognition for multiple categories.

Considerable simplification is possible for the important special case of two-category classification. The two-category decision rule (1), using (14) and (3), simplifies to

$$\left.\begin{aligned} d(X) &= \theta_A \quad \text{if } P(X) > 0 \\ d(X) &= \theta_B \quad \text{if } P(X) < 0 \end{aligned}\right\} \quad (19)$$

where

$$P(X) \triangleq P^A(X) - KP^B(X). \quad (20)$$

Note that in the special case of two-category classification, it is not necessary to use a separate polynomial discriminant function for each category. The function $P(X)$ in (20) is the desired discriminant function given in (10). Just as was desired, $P(X) = 0$ becomes a mathematical description of the decision surface between categories θ_A and θ_B. From (16), (15), and (20) it can be seen that the coefficients of $P(X)$ in (10) can be evaluated by the equations:

$$\begin{aligned}D_{z_1z_2\cdots z_p} = {} & \frac{1}{z_1!z_2!\cdots z_p!\sigma^{2h}} \\ & \cdot\left[\frac{1}{m}\sum_{i=1}^{m} X^{z_1}_{ai1}X^{z_2}_{ai2}\cdots X^{z_p}_{aip}\exp\left(\frac{B_{ai}}{\sigma^2}\right)\right. \\ & \left. - \frac{K}{n}\sum_{i=1}^{n} X^{z_1}_{bi1}X^{z_2}_{bi2}\cdots X^{z_p}_{bip}\exp\left(\frac{B_{bi}}{\sigma^2}\right)\right] \quad (21)\end{aligned}$$

where m = number of training patterns for category θ_A, n = number of training patterns for category θ_B, $h = z_1 + z_2 + \cdots + z_p$, $B_{ai} \triangleq -\frac{1}{2}X^t_{ai}X_{ai}$ and $B_{bi} \triangleq -\frac{1}{2}X^t_{bi}X_{bi}$.

Classification of any point X into one of the two categories θ_A or θ_B can now be accomplished by means of the decision rule (19), where $P(X)$ is defined by (10) and the coefficients of (10) are calculated using (21). This method of classification of points X is the "polynomial discriminant method" (PDM) of pattern recognition for two categories.

One might ask what advantage the PDM approach has over the straightforward formation of the powers and cross products of the original measurement variables X_j and treatment of these derived variables simply as additional inputs to a linear classifier. Sebestyen [15] (pp. 66–74) described a minimum mean-squared-error procedure which finds the optimal coefficients for a polynomial of given order provided that the number of training patterns is very much greater than the number of coefficients to be calculated. For smaller numbers of training samples, however, this classifier tends to overfit the data points. The resulting classifier becomes very accurate in classifying the given training points, but since the procedure involves no estimate of the densities of the underlying distributions, it may be quite poor in classifying patterns not in the training set.

The PDM algorithm was derived in such a way that hundreds or thousands of terms could be introduced into the polynomial discriminant function without overfitting the data even if the number of training points is less than the number of coefficients. It is important to note that actual identity of the polynomial estimator and the estimator of (7) occurs when all possible terms are included in the polynomial. The significant point is that the polynomial approximation tends to fit the estimator of (7), *not the actual data*. This estimator already involves smoothing of the data—thereby minimizing the effects of randomness in sampling on the resulting decision surfaces. Thus the number of terms used is limited only to minimize computation; there is no need to further limit this number because of any danger of overfitting the data. As a practical matter, the computed polynomial can usually be truncated to contain a small number of terms with little degradation in its discriminatory ability.

A second difficulty with the least-squares solution, as pointed out by Sebestyen, is that calculation of the coefficients of a polynomial having N terms requires the storage, computation, and inversion of an $N \times N$ matrix. In contrast, the computational and storage requirements for calculation of coefficients using the PDM grow only linearly with N.

Properties of the PDM

A. General

The two-category PDM classifier represented by (21) of course retains the important feature of the many-category PDM classifier represented by (16), in that all coefficients can be adjusted by adding the effects of each pattern, one pattern at a time. It should be noted that this is not an iterative procedure, as many one-pattern-at-a-time procedures are; one pass through the training data yields the complete solution. Since one access to each pattern is sufficient, it is not necessary to store the training data.

Note that when (21) is used for training, increments due to patterns of category θ_A are multiplied by $1/m$ and added into the coefficients, and increments due to patterns of category θ_B are multiplied by K/n and subtracted from the coefficients. Because the magnitude of $P(X)$ in (10) is not used in the classification decision, $P(X)$ can be multiplied through by a constant with no effect on the decision boundary. Suppose each of the coefficients in $P(X)$ were multiplied by m. Then (21) shows that increments due to category θ_A patterns are simply added into the coefficients, while increments due to category θ_B patterns are multiplied by $K(m/n)$ and subtracted from the coefficients. Therefore, it is not necessary to know the value of either m or n before training can commence, but only their ratio. This fact could have practical value in on-line applications in which the duration of the training phase or the frequency of occurrence of training samples is not known in advance.

One might question whether use of an infinite series such as (10) as a discriminant function could be practical. Truncation of the series is obviously necessary, but how many terms must be retained to avoid sacrificing accuracy of discrimination? The answer to this question depends in part on the behavior of the coefficients as the degree of the corresponding variable becomes large.

An experiment was performed in which 100 patterns were taken from a four-dimensional correlated Gaussian distribution with mean (1, 1, 1, 1) and a correlation matrix which has eigenvalues (1.0, 1.5, 2.0, 1.0). All possible coefficients up to and including tenth order were calculated and printed. For $\sigma = 1.0$, only 87 of the 1025 coefficients calculated had values greater than 5 percent of the maximum value of any coefficient; and of these, none were coefficients of variables greater than the sixth order. For $\sigma = 3.0$, only 5 coefficients had values above 5 percent of the maximum; these were limited to the first-order coefficients and D_0. This experiment, and later experience with actual physical data, provided convincing evidence that a relatively small number of terms are sufficient to define an excellent decision boundary.

From the results just described it is seen that as the smoothing parameter σ is increased from 1 to 3, the order of the decision surface is essentially reduced from sixth order to first order. This can be explained heuristically in the following way: as σ is increased, the estimated density is smoothed and can therefore be represented by lower order terms. On the other hand, as σ gets smaller, the estimated density can approximate the true distribution of the data in more detail. For a given number of training samples, σ should be at least large enough to provide smoothing between adjacent training samples, and may be increased above this minimum to limit the order of the decision boundary and thus the number of terms required in the polynomial.

Analytic solutions to the interrelated problems of truncation, selection of σ, and number of training samples needed are treated in Specht [1] and will be published in a subsequent paper.

B. Consistency of the Density Estimates

It has been pointed out that the accuracy of the decision boundaries is dependent on the accuracy with which the underlying probability densities are estimated. Parzen [17] and Murthy [18], [19] have shown how one may construct a family of estimates of $f(X)$ which include the estimator of (7) and which are consistent (asymptotically approach identity with the probability density function) at all points X at which the density function is continuous, providing $\sigma=\sigma(n)$ is chosen as a function of n such that

$$\lim_{n\to\infty} \sigma(n) = 0 \tag{22}$$

and

$$\lim_{n\to\infty} n\sigma(n) = \infty. \tag{23}$$

The same statement can, of course, be made for the estimated density expressed in polynomial form, (14), since (7) and (14) are equivalent.

C. Limiting Conditions as $\sigma\to 0$ and as $\sigma\to\infty$

Consider the relative contributions of two terms from the estimator (7), corresponding to points X_{a1} and X_{a2}. Let

$$R_{a1} \triangleq (X_{a1}, X)$$

and

$$R_{a2} \triangleq (X_{a2}, X) = R_{a1} + \epsilon,\ \epsilon > 0.$$
$$(X_{ai}, X) \triangleq [(X_{ai} - X)^t(X_{ai} - X)]^{1/2}.$$

The ratio of the contribution of these two points to the value of $f_A(X)$ is

$$\frac{\exp\left[-(R_{a1})^2/2\sigma^2\right]}{\exp\left[-(R_{a1}+\epsilon)^2/2\sigma^2\right]} = \exp\left[\frac{R_{a1}\epsilon + \epsilon^2/2}{\sigma^2}\right]. \tag{24}$$

In the limit as $\sigma\to 0$, this ratio approaches infinity. In other words, as $\sigma\to 0$ the training sample closest to X has infinitely more effect on $f_A(X)$ than any other, and so the others can be ignored. Similarly, in considering the decision rule (4), as $\sigma\to 0$ the closest training point in category θ_r (having distance R_r from X) determines $f_r(X)$, while the closest training point in any other category θ_s (having distance R_s from X) determines $f_s(X)$. Thus the classification of X is determined to be the same as that of the closest training pattern. This limiting case is known as the "nearest-neighbor decision rule," and has been investigated in detail by Cover and Hart [12].

Now consider the opposite limiting condition on σ, i.e., $\sigma\to\infty$. The decision surface from (2), with $K=1$, using the estimator (7) for each category and with exponentials expanded by the Taylor series, is

$$\frac{1}{m}\sum_{i=1}^{m}\left[1 - \frac{(X_{ai}, X)}{\sigma^2} + \frac{(X_{ai}, X)^2}{2!\sigma^4} - \cdots\right]$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left[1 - \frac{(X_{bi}, X)}{\sigma^2} + \frac{(X_{bi}, X)^2}{2!\sigma^4} - \cdots\right]. \tag{25}$$

As $\sigma\to\infty$

$$\frac{1}{m}\sum_{i=1}^{m}(X_{ai} - X)^t(X_{ai} - X)$$
$$= \frac{1}{n}\sum_{i=1}^{n}(X_{bi} - X)^t(X_{bi} - X).$$

Then

$$\frac{1}{m}\sum_{i=1}^{m}(2X^tX_{ai} - X_{ai}{}^tX_{ai}) = \frac{1}{n}\sum_{i=1}^{n}(2X^tX_{bi} - X_{bi}{}^tX_{bi})$$

and

$$2X^t[M_A - M_B] = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{p}X_{aij}^2 - \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}X_{bij}^2. \tag{26}$$

It will now be shown that the decision boundary given by (26) is identical to that obtained using matched filters when the distributions of the categories have equal diagonal elements in their convariance matrices. Define $M_A{}^t=[M_{a1}\cdots M_{aj}\cdots M_{ap}]$. Then from (26),

$$2X^t[M_A - M_B]$$
$$= \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{p}[(X_{aij} - M_{aj})^2 + 2X_{aij}M_{aj} - M_{aj}^2]$$
$$- \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}[(X_{bij} - M_{bj})^2 + 2X_{bij}M_{bj} - M_{bj}^2]$$
$$= M_A{}^tM_A - M_B{}^tM_B. \tag{27}$$

From Nilsson [6], p. 18, it is seen that (27) defines the matched filter (minimum distance classifier) decision boundary. It is interesting to note that whereas the matched filter solution is based solely on the means, the boundary of (26) is sensitive to the diagonal terms in covariance matrices of the distributions as well as to their means.

Since the PDM (10), (19), and (21) were derived directly from (2) and (7) without approximation, they define the same boundaries. Therefore the statements made concerning limiting conditions for the latter apply also to the former. Thus the polynomial discriminant method of pattern classification, which was designed so that a finite amount of smoothing could be applied to distributions of training samples, converges to suboptimal but useful methods even for the extreme cases of $\sigma\to\infty$ and $\sigma\to 0$. Note that the separating boundary

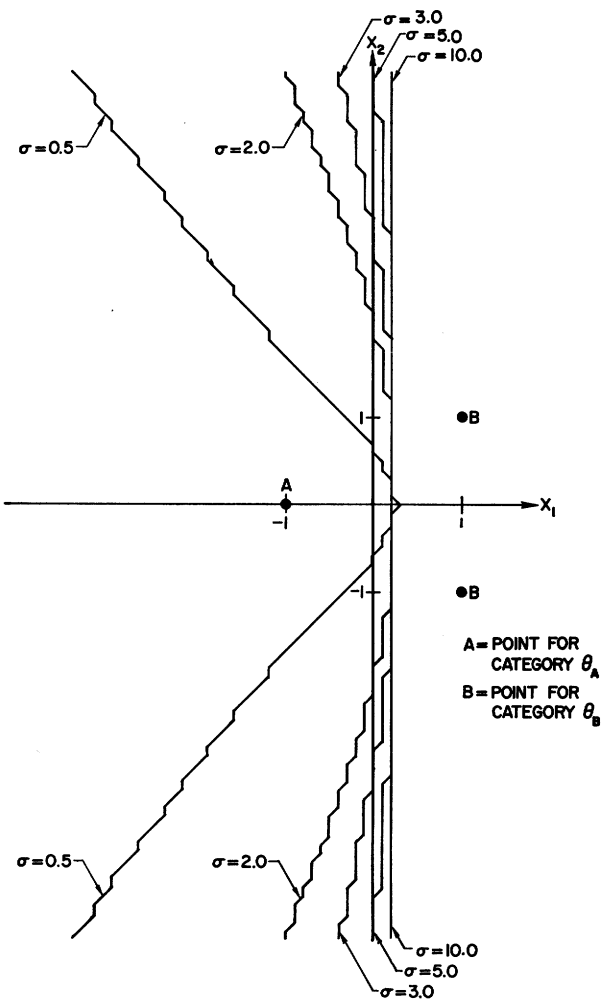

Fig. 3. PDM separation of one point in category θ_A from two points in category θ_B.

ranges from strictly linear for $\sigma \to \infty$ to highly nonlinear for $\sigma \to 0$.

Some Two-Dimensional Examples of the Classification Capabilities of the PDM

The two-category classifier of (10), (19), and (21) has been programmed on both the IBM 7090 and IBM 1620 computers. While the 7090 program handles problems up to 46 dimensions (and 10 000 possible coefficients), the 1620 program is limited to two dimensions, but it has the capability for plotting the training points and the calculated boundary. The boundaries for four simple problems using artificial data were calculated and plotted; they are displayed in Figs. 3, 4, 5, and 6.

Figure 3 shows one point from category θ_A and two points from category θ_B. As σ ranges from 0 to ∞, the boundary varies from the limit consisting of two straight lines ($\sigma \to 0$: nearest neighbor rule) through higher order curves which are approximately parabolic, through lower order curves which are more closely parabolic, to a straight line parallel to the X_2 axis—which is the limit-

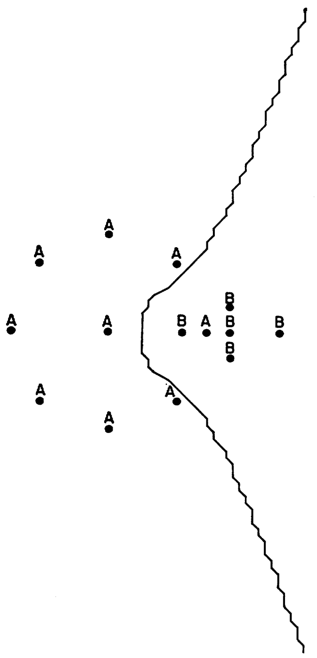

Fig. 4. PDM separation of overlapping distributions ($\sigma = 0.5$).

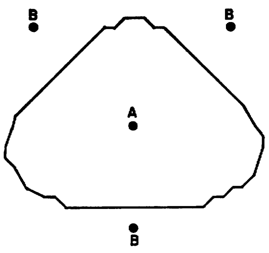

Fig. 5. PDM separating boundary between one point representing category θ_A and 3 points representing $\theta_B (\sigma = 0.5)$.

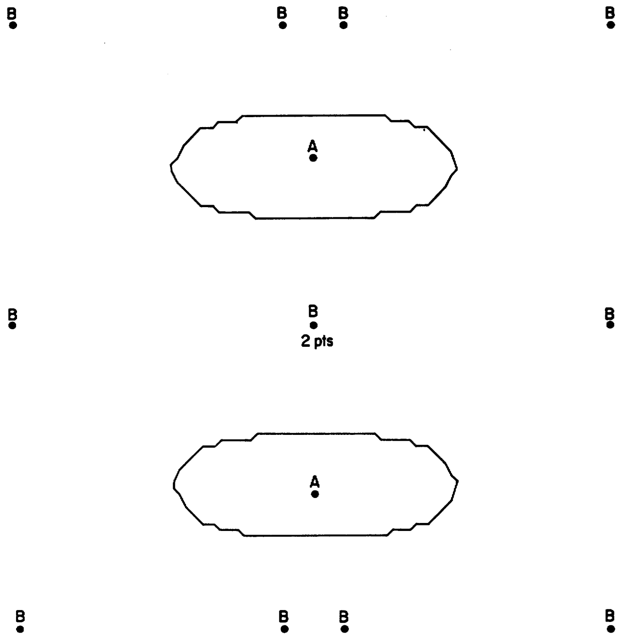

Fig. 6. PDM separating boundary between a bimodal distribution and one which is more nearly uniform ($\sigma = 1.5$).

ing case as $\sigma \to \infty$. The equation $P(X_1, X_2)=0$ determines a boundary which has continuous derivatives; the stairstep effect in the boundaries as plotted can be attributed to coarse resolution in the plot routine.

Figure 4 illustrates the common problem of separation of two overlapping distributions with unequal covariance matrices. It is to be expected that the boundary tends to surround the category with smaller variance because its estimated probability density diminishes much faster with distance from its mean, than the estimated density of the other category diminishes with distance from *its* mean.

Figures 5 and 6 demonstrate the flexibility of the polynomial discriminant method. If one category is surrounded by training points of the other category, the calculated boundary can completely enclose that category—and does so automatically when necessary, as shown in Fig. 5. Figure 6 illustrates the boundary obtained when category θ_A has a bimodal distribution superimposed on a distribution θ_B which is more nearly uniform. The boundary obtained by setting $P(X)=0$ in the PDM not only can completely surround a category, but can isolate two or more closed regions!

Implementation of the PDM

The polynomial discriminant method can be implemented (either off line or on line) by means of programs for general-purpose computers; all of the experimental work reported in this paper was performed in this way. However, whenever a particular pattern-recognition application involves a high volume of data reduction over an extended period of time—particularly if the data reduction must be performed in real time—a special-purpose computer may be justified or even mandatory. It was this need to consider hardware (special-purpose computer) implementation which served as partial motivation for development of a discriminant function of polynomial form, rather than of some other form, because of the relative ease with which polynomials can be mechanized in the form of special-purpose computers and because such a mechanization is a logical extension of the adaptive linear threshold devices (Adalines) described by Widrow and Hoff [20], Koford and Groner [8], and others.

A. Two-Category Classifier

When it is desired to implement a separating surface more general than the linear, it is natural simply to add powers and cross products of the input variables as new (derived) input variables with separate coefficients. The result is the polynomial threshold element of Fig. 7. The coefficients can, of course, be calculated using (21). As indicated, the output of the summer represents $P(X)$ of (10). The quantizer operating on $P(X)$ causes the output of the threshold element to be one of two states indicating the decisions "category θ_A" and "category θ_B" corresponding to the conditions $P(X)>0$ and $P(X)<0$, respectively.

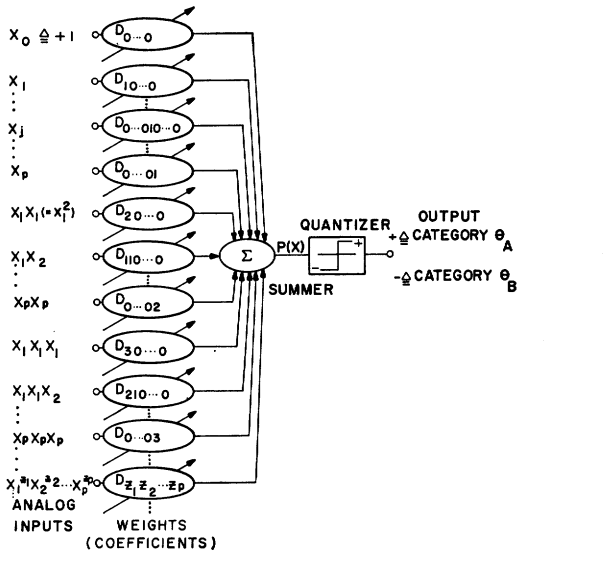

Fig. 7. A polynomial threshold element.

If the statistics of the problem are stationary (i.e., unchanging with time), it may be advisable to collect a representative sample of training patterns with known desired outputs, calculate the coefficients D using a general-purpose digital computer, and then construct a physical threshold device with fixed weights to classify new patterns as they occur. Assuming that the input variables and the cross products of interest can be obtained in the form of voltages, the weights D in Fig. 7 could be mechanized by simple resistors having conductances proportional to the absolute value of the calculated coefficients; the summer by a summing junction; and the quantizer possibly by a high-gain amplifier which normally operates saturated either in the positive direction or in the negative direction—indicating category θ_A or θ_B, respectively. If resistors are used for weights, an inversion of the input signal is necessary for each case in which the calculated weight is negative.

Note that the above mechanization can be used, even if a representative set of training samples is not immediately available or if the statistics of the problem are slowly time-varying, by means of periodic recalculation of coefficients and changing of resistor values if necessary. However, if periodic off-line calculation is not sufficient, either because of the time required to utilize new data or for reasons of operational convenience, then adaptive hardware as indicated in Fig. 8 can be used. Notice that the normalizing factor $\exp(X^tX/2\sigma^2)$ is calculated only once per pattern and is then used with all of the weights during training of the classifier, and need not even be calculated during the test and use phases. The premultiplying factor $(z_1!z_2! \cdots z_p!\sigma^{2h})^{-1}$ is effectively split into two equal factors by using the square root of this factor as the constant of an attenuator applied to the input voltage representing a cross product. In this way, the square root of the premultiplying factor is applied once during training (the latter consists of computation of the coefficients) and again

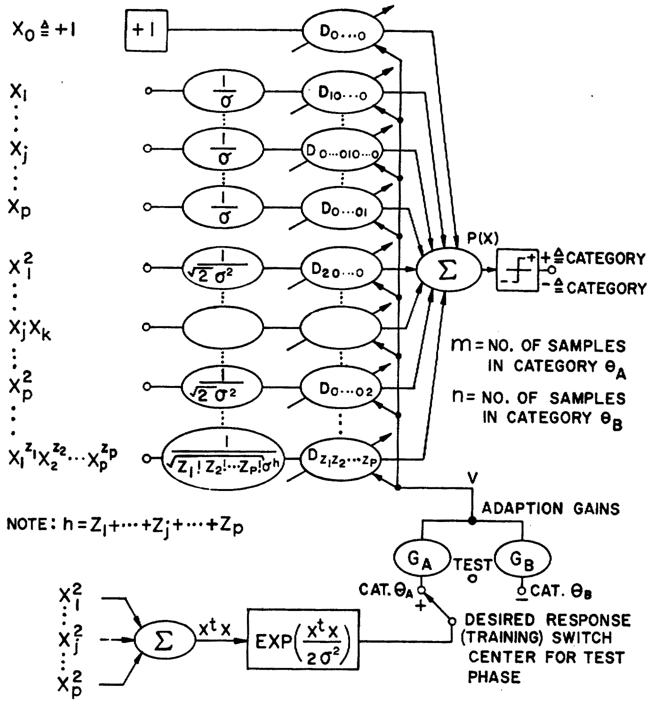
$X_0 \triangleq +1$
+1
X_1
X_j
X_p
X_1^2
X_jX_k
X_p^2
$X_1^{z_1}X_2^{z_2}\cdots X_p^{z_p}$
$\frac{1}{\sigma}$
$\frac{1}{\sqrt{2}\,\sigma^2}$
$\frac{1}{\sqrt{z_1!\,z_2!\cdots z_p!}\,\sigma^h}$
$D_{0\cdots0}$
$D_{10\cdots0}$
$D_{0\cdots010\cdots0}$
$D_{0\cdots01}$
$D_{20\cdots0}$
$D_{0\cdots02}$
$D_{z_1z_2\cdots z_p}$
Σ
P(X)
+ ≙ CATEGORY θ_A
− ≙ CATEGORY θ_B
m = NO. OF SAMPLES IN CATEGORY θ_A
n = NO. OF SAMPLES IN CATEGORY θ_B
V
ADAPTION GAINS
NOTE: $h = Z_1 + \cdots + Z_j + \cdots + Z_p$
G_A
G_B
TEST
CAT. θ_A
CAT. θ_B
+
−
DESIRED RESPONSE (TRAINING) SWITCH CENTER FOR TEST PHASE
X_1^2
X_j^2
X_p^2
X^tX
$\mathrm{EXP}\left(\frac{X^tX}{2\sigma^2}\right)$

Fig. 8. An adaptive polynomial threshold element.

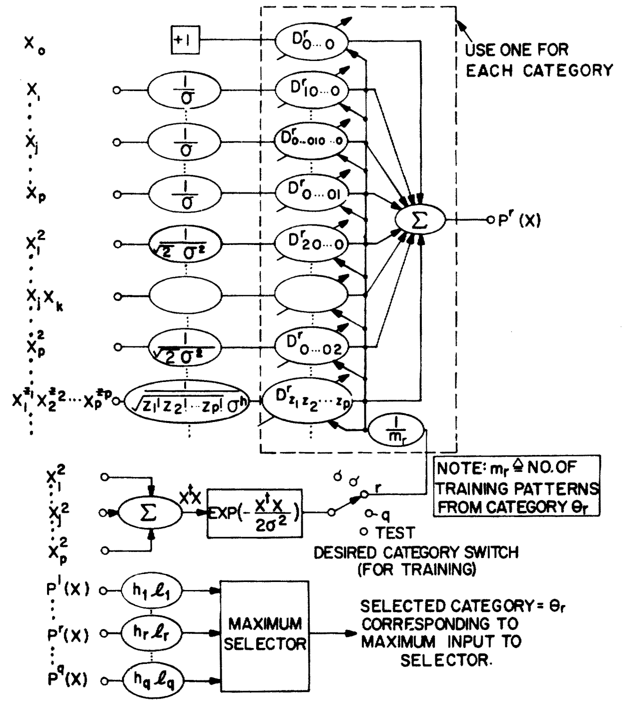
X_0
X_1
X_j
X_p
X_1^2
X_jX_k
X_p^2
$X_1^{z_1}X_2^{z_2}\cdots X_p^{z_p}$
+1
$\frac{1}{\sigma}$
$\frac{1}{\sqrt{2}\,\sigma^2}$
$\frac{1}{\sqrt{z_1!\,z_2!\cdots z_p!}\,\sigma^h}$
$D^r_{0\cdots0}$
$D^r_{10\cdots0}$
$D^r_{0\cdots010\cdots0}$
$D^r_{0\cdots01}$
$D^r_{20\cdots0}$
$D^r_{0\cdots02}$
$D^r_{z_1z_2\cdots z_p}$
USE ONE FOR EACH CATEGORY
Σ
$P^r(X)$
$\frac{1}{m_r}$
NOTE: m_r ≙ NO. OF TRAINING PATTERNS FROM CATEGORY θ_r
X^tX
$\mathrm{EXP}(-\frac{X^tX}{2\sigma^2})$
θ_σ
r
q
TEST
DESIRED CATEGORY SWITCH (FOR TRAINING)
$P^1(X)$
$P^r(X)$
$P^q(X)$
$h_1\ell_1$
$h_r\ell_r$
$h_q\ell_q$
MAXIMUM SELECTOR
SELECTED CATEGORY = θ_r CORRESPONDING TO MAXIMUM INPUT TO SELECTOR.

Fig. 9. Multiple-category classifier utilizing polynomial discriminant functions.

during testing because it modifies the input before it is multiplied by the coefficient. The elements $D_{z_1z_2\cdots z_p}$ must function as integrators during the training phase, but merely as multipliers during the testing phase. The voltage at point V could be used to control the time that the integrators are allowed to integrate their respective inputs, and the sign of the voltage, which is determined by the desired-response switch position, could control inversion of the input voltage. Regardless of the mechanism chosen, the function of each of the elements during training is to accumulate the sum of the products of the voltage at V times the voltage at the input of the element (calculated once per training pattern) to arrive at the weight $D_{z_1z_2\cdots z_p}$; and during testing, simply to form the product of the final weight and the input. Figure 8 is meant to be a functional block diagram only; mechanization can be purely analog, purely digital, or hybrid. In any case, if the adaption gains G_A and G_B are set at $1/m$ and K/n respectively, it is an exact embodiment of the training rule of (21) and of the decision rule (19). The factor K used in Fig. 8 is that defined by (3).

B. Many-Category Classifier

A very similar mechanization can be used for mechanization of a many-category classifier. From (4) and (14) it is seen that the Bayes decision rule is $d(X) = \theta_r$ such that

$$h_rl_rP^r(X) \geqq h_sl_sP^s(X) \quad \text{for all } s \neq r. \qquad (28)$$

Figure 9 represents schematically the mechanization of a many-category classifier. The main difference between this and Fig. 8 is that in Fig. 9 calculation of the coefficients must be performed separately for each category. During the testing phase, a final determination of category is based on finding the largest of several inputs to the maximum-selector, rather than on detecting whether a single output variable is greater or less than zero as is done for the two-category classifier.

C. Adaptive Capability of Polynomial Threshold Elements

1) *Adaptive Capability to Use Information As It Becomes Available:* In the mechanization of Fig. 8 it is assumed that the numbers of training patterns for categories θ_A and θ_B are known before adaption is begun, and that these numbers are m and n, respectively. Suppose that each of the weights $D_{z_1z_2\cdots z_p}$ were multiplied by the constant m; then the summer output $P(X)$ would be multiplied by the constant m for each pattern X. However, since the classification of X depends on the sign of $P(X)$ and not on its magnitude, multiplication of each of the coefficients $D_{z_1z_2\cdots z_p}$ by $m(m>0)$ does not affect the implemented decision surface. This multiplication can be accomplished as a multiplication by m in the adaption gains. This results in $G_A = 1$ and $G_B = -Km/n$. Thus the absolute numbers of training patterns from the two categories need not be known before adaption can be started; only their ratio m/n need be known or estimated.

In many problems, even the ratio m/n need not be known before the one-pattern-at-a-time adaption algorithm can be applied. Whenever it is reasonable to estimate the probability of occurrence of a pattern from category θ_A by the ratio of the number of occurrences of category θ_A training patterns to the total number of training patterns, the estimate of the value of K is

$$\overline{K} = \frac{\dfrac{n}{m+n}}{\dfrac{m}{m+n}} \frac{l_B}{l_A} = \frac{n}{m} \frac{l_B}{l_A}, \tag{29}$$

and the adaption gains of Fig. 8 become $G_A = 1$ and $G_B = -l_B/l_A$ when this estimate is used. Thus, the only time a priori knowledge of the ratio of number of training patterns from the two categories is necessary is when there is reason to estimate the ratio of a priori probabilities of the two categories by something other than the ratio of frequencies of occurrence of training samples from the two categories. This characteristic makes possible utilization of the mechanization of Fig. 8 in real-time applications because in this mechanization the decision surface is continuously based on all previous information and, as explained above, no lack of accuracy results from lack of foreknowledge of numbers of training samples or the relative probability of their occurrence.

2) *Adaptive Capability to Follow Nonstationary Statistics:* In the preceding section, an adaptive capability in the sense of continuous modification of the decision boundary on the basis of new information was described. This is a desirable approach if the statistics of the problem are stationary and modification of the decision boundary is required in order to utilize properly a body of data which is increasing in size with time. However, if the statistics of the problem are *changing* with time, it would be well to eliminate the influence of earlier samples on the decision boundary and base the latter only on the more recent training samples. Although the dynamics of a particular problem govern the relative weights that should be applied to old training samples in a time-varying situation, a particularly convenient weighting function is the exponential:

$$\text{Weight applied to a sample } (\mu + 1) \text{ samples old} \sim \exp\left(-\frac{\mu}{\tau}\right), \tag{30}$$

where $\mu = 0$ is the age of the present sample, and τ is the parameter ("time constant") of the weighting function —measured in samples rather than in units of time.

Let us consider again the problem of estimation of a density from a finite number of samples. In the time-invariant case, estimation of the density based on m samples from a given category was performed in two steps: first the density of the samples was represented by impulses of magnitude $1/m$ at the locations of the sample points, and then the impulse representation was convolved with a spatial filter whose impulse response is given by (8) in order to obtain a smooth representation of the density. In the present case, the weighting function (30) requires that the density of the samples be estimated by convolution of $h(X)$ with impulses of magnitude $1/\tau \exp(-\mu/\tau)$ at the locations of the sample points, instead of with impulses having the uniform magnitude $1/m$. Thus, when a new training sample is observed, it is represented by an impulse of a probability density whose magnitude is initially $1/\tau$ but which decreases exponentially to zero as newer training samples are observed.

Fortunately, the weighting function (30) can be incorporated into the training algorithm associated with the classifier of Fig. 9 with relatively little change. The training algorithm for the stationary case stated in the form of a recursion equation is

$$\begin{aligned} D^A_{z_1 z_2 \cdots z_p}(i) &= D^A_{z_1 z_2 \cdots z_p}(i-1) \\ &\quad + \frac{1}{z_1! z_2! \cdots z_p! m\sigma^{2h}} X^{z_1}_{ai1} X^{z_2}_{ai2} \cdots X^{z_p}_{aip} \\ &\quad \cdot \exp\left(\frac{B_{ai}}{\sigma^2}\right), \end{aligned} \tag{31}$$

where $D^A{}_{z_1 z_2 \cdots z_p}(i)$ is the partial summation of (16) after using i out of m training patterns. Since it is well known from numerical analysis that a recursion equation of the form

$$Y_o(i) = \frac{\tau - 1}{\tau} Y_0(i-1) + \frac{1}{\tau} Y_{\text{in}}(i) \tag{32}$$

represents the sampled-data equivalent of a filter with input Y_{in}, output Y_o, and exponential impulse response, it is obvious that in order to apply the exponential weighting function (30) to training samples (inputs), it is necessary only to add the factor $(\tau - 1)/\tau$ to (31), yielding

$$\begin{aligned} D^A_{z_1 z_2 \cdots z_p}(i) &= \frac{\tau - 1}{\tau} D^A_{z_1 z_2 \cdots z_p}(i-1) \\ &\quad + \frac{1}{z_1! z_2! \cdots z_p! \tau\sigma^{2h}} X^{z_1}_{ai1} X^{z_2}_{ai2} \cdots X^{z_p}_{aip} \\ &\quad \cdot \exp\left(\frac{B_{ai}}{\sigma^2}\right). \end{aligned} \tag{33}$$

Here τ is the "time constant" of the weighting function (in samples rather than actually in units of time) and has no functional relationship to the total number of training patterns used by the classifier. Many other weighting functions besides the exponential can be approximated by somewhat more complicated recursion equations than (32). These too could be used to obtain training algorithms, similar to (33), which are useful for problems with nonstationary statistics.

Conclusions

It has been shown that the polynomial discriminant method (PDM) of pattern recognition developed in this paper possesses the following important features.

1) It provides a simple method of determining weights for cross-product and power terms in the vari-

able inputs to an adaptive threshold element used for statistical pattern classification. The calculation for the coefficient of a particular cross product amounts to little more than averaging that cross product over all the training patterns.

2) The algorithms developed adjust the coefficients of the polynomial (the weights) on a one-pattern-at-a-time basis. The procedure is not iterative; learning is complete after each pattern has been observed only once. These features make storage of training patterns unnecessary and eliminate considerations of convergence rates necessary with other one-pattern-at-a-time algorithms.

3) Since coefficients are adjusted after each pattern, the PDM is able to use new information as it becomes available.

4) With minor modification to the basic adapt algorithm, the PDM is able to disregard old data and therefore to follow nonstationary statistics (still without need to store old training patterns explicitly).

5) The PDM decision surfaces can asymptotically approach Bayes-optimal decision surfaces for all regions for which the probability density functions of the categories to be separated are continuous.

6) The shape of the decision surfaces can be made as complex as necessary, or as simple as desired, by proper choice of the smoothing parameter σ.

7) The computational and storage requirements of the PDM increase only linearly with the number of coefficients used.

8) Because of the smoothing properties inherent in the PDM, the number of coefficients used can approach or even exceed the number of training patterns with no danger of the polynomial overfitting the data.

Practical implementation of the PDM, both in the form of special-purpose computer hardware (fixed and adaptive) and in the form of programs for general purpose computers, is described. The interrelated problems of selection of values for σ, of degree of truncation for the polynomials, and of the number of training samples necessary for practical classification problems are treated extensively in Specht [1]. It is expected that this material will be published as a separate paper which will also contain the derivation of a simplified version of the PDM for the restricted case of binary input variables.

The polynomial discriminant method has been applied quite successfully to the problem of automatic analysis of electrocardiograms [16]. The results reported in that paper are of interest not only because of the remarkably high diagnostic accuracy of the PDM, but also because they show, in one practical problem at least, that useful polynomial discriminant functions can contain a relatively small number of terms even though an infinite number are possible. In this investigation it was found that a polynomial with 30 coefficients was sufficient to classify points in a 46-dimensional measurement space, and that it did so with the same accuracy on cases not in the training set as did a classifier using the original estimator of (7). It was also demonstrated in this example (and has been noted by the author in the solution of other practical problems) that although the best value of σ is dependent on the statistical nature of the data, peak accuracy can be obtained over a wide enough range that selection of a good value of σ is not at all difficult.

Acknowledgment

The author wishes to express his gratitude to Dr. Bernard Widrow for his continued support and advice, and to Dr. C. S. Weaver, K. A. Belser, L. R. Talbert and others at Stanford University for many stimulating discussions.

References

[1] D. F. Specht, "Generation of polynomial discriminant functions for pattern recognition," Ph.D. dissertation, Stanford University, June 1966 [also available as Rept. SU-SEL-66-029 (TR 6764-5), Stanford Electronics Labs., Stanford, Calif., May 1966 and as Defense Documentation Center Rept. AD 487 537].

[2] B. Widrow, G. F. Groner, M. J. C. Hu, F. W. Smith, D. F. Specht, and L. R. Talbert, "Practical applications for adaptive data-processing systems," WESCON Technical Paper 11.4, August 1963.

[3] B. Widrow and F. W. Smith, "Pattern-recognizing control systems," in *Computer and Information Sciences*, Julius Tou and R. H. Wilcox, Eds. Washington, D. C.: Spartan Books, 1964, pp. 288–317.

[4] B. Widrow, "Adaptive sampled-data systems—a statistical theory of adaption," *1959 WESCON Conv. Rec.*, pt. 4, pp. 74–85.

[5] F. Rosenblatt, *Principles of Neurodynamics.* Washington, D. C.: Spartan Books, 1962.

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

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

[8] J. S. Koford and G. F. Groner, "The use of an adaptive threshold element to design a linear optimal pattern classifier," *IEEE Trans. on Information Theory*, vol. IT-12, pp. 42–50, January 1966.

[9] T. Kailath, "Adaptive matched filters," in *Mathematical Optimization Techniques*, R. Bellman, Ed. Berkeley, Calif.: University of California Press, 1963, pp. 109–140.

[10] K. Steinbuch and U. A. W. Piske, "Learning matrices and their applications," *IEEE Trans. on Electronic Computers*, vol. EC-12, pp. 846–862, December 1963.

[11] B. Widrow, "Generalization and information storage in networks of Adaline 'neurons'," in *Self-Organizing Systems—1962*, M. C. Yovits, G. T. Jacobi, and G. D. Goldstein, Eds. Washington, D. C.: Spartan, pp. 435–461.

[12] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. on Information Theory*, vol. IT-13, pp. 21–27, January 1967.

[13] G. S. Sebestyen, "Pattern recognition by an adaptive process of sample set construction," *IRE Trans. on Information Theory*, vol. IT-8, pp. S82–S91, September 1962.

[14] A. M. Mood and F. A. Graybill, *Introduction to the Theory of Statistics.* New York: McGraw-Hill, 1963.

[15] G. Sebestyen, *Decision-Making Processes in Pattern Recognition.* New York: Macmillan, 1962.

[16] D. F. Specht, "Vectorcardiographic diagnosis using the polynomial discriminant method of pattern recognition," *IEEE Trans. on Bio-Medical Engineering*, vol. BME-14, pp. 90–95, April 1967.

[17] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, pp. 1065–1076, September 1962.

[18] V. K. Murthy, "Estimation of probability density," *Ann. Math. Stat.*, vol. 36, pp. 1027–1031, June 1965.

[19] —— "Nonparametric estimation of multivariate densities with applications," Douglas Paper 3490, Douglas Missile and Space Systems Division, Douglas Aircraft Co., Santa Monica, Calif., June 1965.

[20] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *1960 WESCON Conv. Rec.*, pt. 4, pp. 96–104.