SEQUENTIAL PATTERN RECOGNITION

A Thesis

Presented to

the Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Walter Manning, Jr.

May 1971

584008

SEQUENTIAL PATTERN RECOGNITION

An Abstract of a Thesis

Presented to

the Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Walter Manning, Jr.

May 1971

ABSTRACT

Consider an n-dimensional space which has been partitioned into "t" unique subspaces, called categories or populations. Associated with each category is a set of n-tuples, to be referred to as pattern vectors. Since each pattern vector belongs to one and only one category, each vector may be considered to be a data vector having "n" dimensions and belonging to a specific category.

A program was written which realizes an algorithm that performs an adaptive process with data of known classification. This procedure will establish the necessary criterion for a classification scheme for other data from the same space whose category is unknown. A second program was written which performs the classification process according to the criterion established by the adaptive technique. It should be noted that a priori knowledge about the probability distributions of the data sets need not be known.

Several test problems were run on the IBM 360 digital computer using geophysical data. The results of these runs were highly successful in correctly classifying the data that was used.

TABLE OF CONTENTS

CHAPTER I

PATTERN RECOGNITION

1.1 Introduction

Suppose that an n-dimensional space is partitioned into "t" subspaces, called categories or populations. Each point in this space is considered to be an n-dimensional vector, a pattern vector. Each category may have an infinite number of pattern vectors associated with it. It is desirable to develop a model which will, according to some given algorithm, classify each pattern vector to its appropriate category.

As an example, suppose that it is desirable to distinguish between earthquakes originating from different depths (a problem to be considered later in this paper). The categories could be chosen according to arbitrarily selected depths of occurrance, whereas the data taken from a seismograph would determine the number of dimensions of each pattern vector.

1.2 Discriminant Functions

A pattern vector, say X, is a point in an n-dimensional space. The coordinates which describe this vector $(x_1\ x_2\ \ldots\ x_n)$ are the real numbers. A pattern classifier can therefore be considered as an instrument which maps

the points in the n-dimensional space into the subspace or category numbers 1, 2, . . ., t. The boundaries that separate the subspaces are called decision surfaces. Any decision surface can be clearly defined by a set or family of functions called discriminant functions. These functions are chosen such that for all $X \in t_j$, $f_j(X) > f_k(X)$ for j, $k = 1, 2, \ldots t$, $j \neq k$, where f_j and f_k are single-valued scalar functions of the pattern vector X, and t_j is one of the "t" partitioned subspaces. Extending this concept, consider the discriminant function

$$\mu_{jk}(X) = f_j - f_k \tag{1.1}$$

then

$$\mu_{jk} \geq o \text{ iff } f_j \geq f_k \tag{1.2}$$
$$\mu_{jk} < o \text{ iff } f_j < f_k$$

hence

$$X \in t_j \text{ iff } \mu_{jk} \geq o \tag{1.3}$$
$$X \in t_k \text{ iff } \mu_{jk} < o$$

The choice of the discriminant function is a compromise between two considerations: (1) the desired accuracy of classification, and (2) the economics considerations.

For example, a linear discriminant function will not generally partition the space as precisely as would a quadratic function; however, the linear function will be the more economical of the two in terms of the necessary computation and storage required. Consequently, a "trade-off" situation develops. The solution to this is to consider the characteristics of the particular problem to be solved.

Discriminant functions may be of any order; however, the most commonly used are the linear and the quadratic forms. Equation (4) below is the general linear discriminant function using three variables, and Equation (5) is the quadratic form using the same variables.

$$\mu_1(x,y,z) = a_1x + a_2y + a_3z + a_4 \tag{1.4}$$

$$\mu_2(x,y,z) = a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5xz + a_6yz + a_7x + a_8y + a_9z + a_{10}$$

Once the discriminant function has been chosen, the problem then becomes a matter of choosing the "best" values for the coefficients $a_1, a_2, \ldots, a_m$. The vector A, denoted by $A = (a_1 \ a_2 \ \ldots \ a_m)$, is called the weight vector and the vector components are called the weights. It should be noted that if there are more than two categories, then

additional discriminant functions will be necessary to perform the pattern classification.

1.3 Bayes Discriminant Function

The classification problem has been solved by statistical decision theory if the probability density functions are known for each population. The discriminant function which performs this "optimum" decision process is known as the Bayes discriminant function.

Minimizing the probability of misclassification is the most desirable trait for any classifier. The central issue of the decision-theoretic treatment is the specification of a loss function, $C(j/k)$, $j, k = 1, 2, \ldots, t$, where t equals the total number of unique categories. Consider a two-category problem, t_1 and t_2. There are only two types of errors that can be made: (1) classifying an observation as coming from t_1 when it belongs to t_2, designated $C(1/2)$, and (2) classifying an observation as coming from t_2 when it belongs to t_1, designated $C(2/1)$.

Let the probability that an observation will come from t_1 be q_1 and from t_2 be q_2. Let the probability density function for t_1 be $p_1(X)$ and for t_2 be $p_2(X)$, where $X = (x_1\ x_2\ \ldots\ x_n)$. From Bayes Theorem, the conditional probability that, given an observation X, it comes from population t_i is

$$P(t_i/X) = \frac{q_i p_i(X)}{q_1 p_1(X) + q_2 p_2(X)} \tag{1.6}$$

If X is classified as belonging to t_1, then the expected loss, also called a conditional average loss, for misclassification is

$$L_1(X) = C(1/2)P(t_2/X) \tag{1.7}$$

Using Equation (1.6) yields

$$L_1(X) = \frac{C(1/2)q_2 p_2(X)}{q_1 p_1(X) + q_2 p_2(X)} \tag{1.8}$$

Similarly, if X is classified as belonging to t_2,

$$L_2(X) = \frac{C(2/1)q_1 p_1(X)}{q_1 p_1(X) + q_2 p_2(X)} \tag{1.9}$$

The minimum expected loss is obtained by assigning each X to the population which has the smallest expected loss. This rule is called the Bayes Decision Rule, and is expressed as

$$\begin{aligned} D(X) &= L_2(X) - L_1(X) \\ &= \frac{C(2/1)q_1 p_1 - C(1/2)q_2 p_2}{q_1 p_1 + q_2 p_2} \end{aligned} \tag{1.10}$$

where

$$X \in t_1 \text{ iff } D(X) \geq o \tag{1.11}$$

$$X \in t_2 \text{ iff } D(X) < o$$

The function D(X) is called the Bayes discriminant function.

When there are more than two populations with known probability densities, the procedure is to, as before, minimize the total expected loss. For example, consider a three-category problem. Let C(j/i) be the cost of misclassifying an observation X as coming from t_j when it belongs to t_i. The resulting expected losses are:

$$L_1(X) = \frac{C(1/2)q_2p_2 + C(1/3)q_3p_3}{\sum_{n=1}^{3} q_np_n} \tag{1.12}$$

$$L_2(X) = \frac{C(2/1)q_1p_1 + C(2/3)q_3p_3}{\sum_{n=1}^{3} q_np_n} \tag{1.13}$$

$$L_3(X) = \frac{C(3/1)q_1p_1 + C(3/2)q_2p_2}{\sum_{n=1}^{3} q_np_n} \tag{1.14}$$

The discriminant functions are defined as:

$$D_{ij}(X) = L_j(X) - L_i(X) \tag{1.15}$$

which results in

$$D_{12}(X) = L_2(X) - L_1(X) \tag{1.16}$$

$$D_{13}(X) = L_3(X) - L_1(X) \tag{1.17}$$

$$D_{23}(X) = L_3(X) - L_2(X) \tag{1.18}$$

The decision rules are

$$X \in t_1 \text{ iff } D_{12} \geq o \text{ and } D_{13} \geq o \tag{1.19}$$

$$X \in t_2 \text{ iff } D_{12} < o \text{ and } D_{23} \geq o$$

$$X \in t_3 \text{ iff } D_{13} < o \text{ and } D_{23} < o$$

Now, consider the case where

$$C(2/1) = C(3/1) \tag{1.20}$$

$$C(1/2) = C(3/2)$$

$$C(1/3) = C(2/3)$$

Then, for $i \neq j \neq k$

$$D_{ij}(X) = \frac{C(j/k)q_k p_k + C(j/i)q_i p_i}{\sum_{n=1}^{3} q_n p_n} - \frac{C(i/k)q_k p_k + C(i/j)q_j p_j}{\sum_{n=1}^{3} q_n p_n} \tag{1.21}$$

Since $C(j/k) = C(i/k)$,

$$D_{ij}(X) = \frac{C(j/i)q_i p_i - C(i/j)q_j p_j}{\sum_{n=1}^{3} q_n p_n} \tag{1.22}$$

Since the decision rule indicates that only the sign is im-

portant when classifying, Equation (1.22) becomes

$$D_{ij}(X) = \frac{C(j/i)q_i p_i - C(i/j)q_j p_j}{q_i p_i + q_j p_j} \qquad (1.23)$$

There are two points worth mentioning at this time. Since the log function is a monotonically increasing function of its argument, taking the logarithm of D(X) results in the same decision process. Also, if the probability density functions are gaussian, then the resulting discriminant functions will be quadratic [Ref. 4]. Generally, the probability density functions for the populations will not be known; hence, the Bayes function can not be applied.

1.4 Adaptive Pattern Recognition

An adaptive pattern classification scheme is one which is able to monitor its own performance, thus enabling it to alter, if necessary, its decision making process according to some stipulated criterion; it is, therefore, considered to be self-optimizing. Central to the adaptive scheme is the "training process", which enables the classifier to achieve the desired performance level. The training proceeds as follows: A set of pattern vectors of known classification, the training set, is used to determine the "optimum" coefficients of the discriminant functions. The system's performance to these coefficients is measured; and,

if the desired performance has been obtained, that discriminant function is used during the classification procedure.

There are at least three significant reasons why an adaptive pattern classifier is used:

(1) This approach offers a reasonable way of processing large quantities of data, where little, if any, information is known concerning the probability distributions of the populations.

(2) An adaptive technique is capable of handling a system whose parameters may fluctuate from time to time.

(3) The adaptive approach can be used to determine the significance of the training set parameters.

There are four desirable characteristics in any adaptive pattern classifier.

(1) The expected loss of the classified patterns should be as close as possible to the minimum expected loss.

(2) The system should possess enough versatility to be capable of handling a broad spectrum of pattern classification problems.

(3) The use of a large-scale digital computer to implement the algorithm would be undesirable.

(4) It is desirable in adaptive systems to express the usefulness of a discriminant function quantitatively, i.e., to measure the system's capability to correctly classify a

pattern vector of an unknown population.

1.5 Summary

In conclusion, this first chapter has outlined four points: (1) The definitions for a pattern vector; category; training set, and adaptive classification scheme have been stated; (2) The use of discriminant functions for establishing decision rules has been discussed; (3) The optimum decision rule-Bayes function-has been derived, and (4) The usefulness as well as the desirable characteristics of an adaptive pattern classification scheme have been presented.

CHAPTER II

AN ADAPTIVE TECHNIQUE

2.1 Patterson's Method

Chapter I discussed pattern recognition in general and also presented a brief introduction regarding adaptive pattern recognition; this chapter will present the development of a particular adaptive pattern recognition scheme.

In August, 1966, Patterson [Ref. 5] presented an adaptive pattern classification scheme. The discriminant functions were of the following form

$$\mu(A,X) = \sum_{i=1}^{m} a_i \phi_i(X) \tag{2.1}$$

where $a_1, a_2, \ldots, a_m$ are the weighting parameters which are adjusted during the adaption process. The $\phi_i(X)$ functions are the scalar functions of the pattern vector which are determined by the choice of the discriminant function. In vector notation

$$\mu(A,X) = A^T \phi(X) \tag{2.2}$$

where

$$A^T = [a_1 \; a_2 \; \ldots \; a_m]^T \tag{2.3}$$

$$\Phi(X) = [\phi_1(X)\phi_2(X) \ . \ . \ . \ \phi_m(X)] \tag{2.4}$$

$$X = (x_1 \ x_2 \ . \ . \ . \ x_d) \tag{2.5}$$

and where

m = the total number of terms in the discriminant function

d = the dimension of the space.

The decision rule was

$$X \ \varepsilon \ t_1 \text{ iff } \mu(A,X) \geq 0 \tag{2.6}$$

$$X \ \varepsilon \ t_2 \text{ iff } \mu(A,X) < 0$$

The performance criterion was defined as

$$M_{N_1N_2}(A) = \frac{q_1}{N_1} \sum_{\alpha=1}^{N_1} [\mu(A,X_\alpha^{(1)}) - C(2/1)]^2 + \frac{q_2}{N_2} \sum_{\beta=1}^{N_2} [\mu(A,X_\beta^{(2)}) + C(1/2)]^2 \tag{2.7}$$

where C(i/j) is the cost for misclassifying a vector X as belonging to t_i when it actually belongs to t_j. The superscripts indicate the category to which the pattern vector belongs, while N_1 and N_2 are the total number of training samples in classes t_1 and t_2 respectively. The criterion

$M_{N_1N_2}$ has two characteristics which make it very attractive:

(1) $M_{N_1N_2}$ is convex and thus lends itself readily to optimization techniques.

(2) $M_{N_1N_2}$ approaches the Bayes discriminant function as N_1 and N_2 become large.

A search technique was used to optimize the performance criterion.

The aforementioned algorithm can be extended to handle a multicategory problem; however, it is not the most desirable approach for the following reason. Suppose that a training set has adapted the system. Furthermore, suppose that one acquires an additional training set. If the system is to be updated using the previous algorithm, i.e., using the second training set to re-adapt the system, then all training samples must be used in the updating process. It is to this problem that Pitt [Ref. 6] directed his attention. The solution was to sequentialize the adaptive routine; hence, once a training vector was used in the adaptive process, it need never be used again, even if an updating process takes place. This sequentialization procedure results in less computer time.

2.2 The Multicategory Problem

Consider the discriminant function $\mu_{jk}(X,A^{jk})$, which discriminates between categories t_j and t_k, where

$$\mu_{jk}(X,A^{jk}) = -\mu_{kj}(X,A^{jk}) \tag{2.8}$$

The A^{jk}'s are chosen such that MN_{jk} is minimized, where

$$MN_{jk} = \frac{1}{N} \sum_{\ell=1}^{T} \sum_{i_\ell=1}^{N_\ell} [\mu_{jk}(X,A^{jk}) - r(j/k)]^2 \tag{2.9}$$

T = the total number of categories

N_ℓ = the total number of samples in category "ℓ"

N = the total number of samples

and

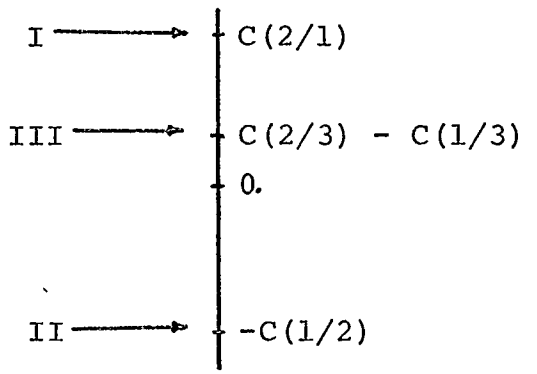
$$\begin{aligned} r(j/k) &= C(k/\ell) - C(j/\ell) && \text{if } \ell \neq j, k \\ &= C(k/\ell) && \text{if } \ell = j \\ &= -C(j/k) && \text{if } \ell = k \end{aligned} \tag{2.10}$$

Then, a sample X_i is said to belong to category t_j if and only if $\mu_{jk}(X_i, A^{jk}) > 0$ for all $j \neq k$.

Suppose that there are three categories, say, 1, 2, and 3. Then

$$\begin{aligned} MN_{12} = \frac{1}{N} \{ & \sum_{X\varepsilon 1} [\mu_{12} - C(2/1)]^2 + \\ & \sum_{X\varepsilon 2} [\mu_{12} + C(1/2)]^2 + \\ & \sum_{X\varepsilon 3} [\mu_{12} - (C(2/3) - C(1/3))]^2 \} \end{aligned}$$

The diagram below will be useful in the following discussion.



As can be seen in the diagram, Category I samples are "mapped" to the point C(2/1), Category II, to -C(1/2) and Category III, to C(2/3) - C(1/3). Intuitively, this mapping is satisfying. Increasing C(2/1) has the effect of increasing the penalty of putting an $X \varepsilon t_1$ into t_2; therefore, it is desirable to map $X \varepsilon t_1$ further from the point 0. Similar reasoning follows for $X \varepsilon t_2$. Now suppose that $X \varepsilon t_3$ and that C(2/3) > C(1/3). Then the cost of putting a sample from t_3 into t_2 is greater than putting it into t_1. So, if a mistake in classification is made, it is better to put the sample in t_1. Using the same reasoning, if C(1/3) > C(2/3), then if a mistake must be made for $X \varepsilon t_3$, map X to t_2.

2.3 The Sequential Approach

Let the (mx1) vector $\Phi(X)$ be defined as

$$\Phi(X) = [\phi_1(X)\phi_2(X) \ . \ . \ . \ \phi_m(X)]^T \qquad (2.12)$$

where "T" means transpose. The $\phi_i(X)$, $i = 1, 2, \ldots, m$, are chosen such that they are linearly independent with probability 1. Define the set of discriminant functions as

$$\mu_{jk}(A^{jk},X) = \sum_{i=1}^{m} a_i^{jk}\phi_i(X)$$

$$= (A^{jk})^T \Phi(X) \qquad (2.13)$$

where $j, k = 1, 2, \ldots t$ such that $1 \leq j < k \leq t$ and where $A^{jk} \varepsilon S^m$. The decision rule is

$$X \varepsilon t_j \text{ iff } \mu_{jk} \geq 0$$

$$X \varepsilon t_k \text{ iff } \mu_{jk} < 0 \qquad (2.14)$$

The set of A vectors are determined sequentially using the training set. Let the (mxm) random matrix P_n be defined as

$$P_n = P_{n-1} + \Phi(n)\Phi^T(n), \quad P_o = \varepsilon I \qquad (2.15)$$

where $\Phi(n) = \Phi(X_n)$, $n = 1, 2, \ldots N$ (N = the total number of training samples), and I is an (mxm) identity matrix. The parameter ε is an arbitrarily small constant such that $\varepsilon > o$. The matrix P_n^{-1} is converted sequentially by the recurrence

relation

$$P_n^{-1} = P_{n-1}^{-1} - [\Phi^T(n) P_{n-1}^{-1} \Phi(n) + 1]^{-1} P_{n-1}^{-1} \Phi(n) \Phi^T(n) P_{n-1}^{-1} \tag{2.16}$$

for n = 1, 2, . . ., N and $P_o^{-1} = \frac{1}{\varepsilon} I$.

Let

$$Z^{jk}(n) = r_\ell(j/k) \text{ iff } X_n \varepsilon t_\ell \tag{2.17}$$

where $r_\ell(j/k)$ is defined by Equation (2.10), which appears below:

$$\begin{aligned} r_\ell(j/k) &= C(k/\ell) - C(j/\ell); \quad \ell \neq j, k \\ &= C(k/j) \quad ; \quad \ell = j \\ &= -C(j/k) \quad ; \quad \ell = k \end{aligned} \tag{2.10}$$

where

$$C(j/k) = 0, \quad j = k \tag{2.18}$$

Let the (mxl) random vector θ_n^{jk} be defined as

$$\theta_n^{jk} = \theta_{n-1}^{jk} + Z^{jk}(n) \Phi(n) \tag{2.19}$$

for n = 1, 2, . . ., N and where θ_o^{jk} is an (mxl) vector whose elements are all zero. After N training samples, the A^{jk} vectors are determined by

$$A_N^{jk} = P_N^{-1}\ \Theta_N^{jk} \tag{2.20}$$

This relation, (2.20), will be proven below.

Equations (2.16) and (2.19) are used to update the system.

2.4 The Performance Criterion

The performance criterion for the sequential scheme is derived from equation (2.7). Let $q_1 = \frac{N_1}{N}$ and $q_2 = \frac{N_2}{N}$; therefore, equation (2.7) becomes, after combining terms,

$$M_{N_1N_2} \triangleq M_N^{jk}(A_N^{jk}) = \frac{1}{N} \sum_{\ell=1}^{t} \sum_{i_\ell=1}^{N_\ell} [\mu_{jk}(A_N^{jk}, X_{i_\ell}) - r_\ell(j/k)]^2 \tag{2.21}$$

where $N_1, N_2, \ldots, N_t$ indicate the number of training vectors in categories 1, 2, . . ., t respectively such that $N_1 + N_2 + \ldots + N_t = N$, and where $\sum_{i_\ell=1}^{N_\ell}$ means that the summation is over all $X_i \in t_\ell$, $i = 1, 2, \ldots, N_\ell$. (The smaller the value of M_N^{jk}, the better the classification.) Now define

$$Q_N^{jk}(A_N^{jk}) = M_N^{jk}(A_N^{jk}) + \frac{1}{N}\, \varepsilon (A_N^{jk})^T\, A_N^{jk} \tag{2.22}$$

It is easily seen that Q_N^{jk} is a continuous, convex function of A_N^{jk}; therefore, the A_N^{jk} vectors which minimize Q_N^{jk} will satisfy

$$\frac{\partial Q_N^{jk}(A_N^{jk})}{\partial A_N^{jk}} = 0 \tag{2.23}$$

where

$$\frac{\partial f(A_N^{jk})}{\partial (A_N^{jk})} = \frac{\partial f(A_N^{jk})}{\partial a_1^{jk}} \cdot \cdot \cdot \frac{\partial f(A_N^{jk})}{\partial a_m^{jk}} \tag{2.24}$$

Performing the indicated operations yields, after simplification,

$$\sum_{\ell=1}^{t} \sum_{i_\ell=1}^{N_\ell} [\Phi(i_\ell)\Phi^T(i_\ell)+\varepsilon I]A_N^{jk} = \sum_{\ell=1}^{t} \sum_{i_\ell=1}^{N_\ell} r_\ell(j/k)\Phi(i_\ell) \tag{2.25}$$

or

$$\sum_{i=1}^{N} [\Phi(i)\Phi^T(i)+\varepsilon I]A_N^{jk} = \sum_{\ell=1}^{t} \sum_{i_\ell=1}^{N_\ell} r_\ell(j/k)\Phi(i_\ell) \tag{2.26}$$

Using equations (2.15) and (2.19), it is easily seen that equation (2.26) becomes

$$P_N A_N^{jk} = \Theta_N^{jk} \tag{2.27}$$

and equation (2.20) follows.

2.5 Setting a Bound for ε

A "poor" initial choice for the value of ε may result in an obviously useless set of A_N^{jk} vectors as indicated by the

magnitude of the performance criterion, M_N^{jk}. The reason for this useless result is round-off error in the computer. Consider the term $[\Phi^T(n)P_{n-1}^{-1}\Phi(n)+1]^{-1}$ in equation (2.16). For convenience, choose

$$\mu_{jk} = a_1x_1 + a_2x_2 + a_3 \tag{2.28}$$

After substituting the appropriate terms and performing the indicated matrix operations, one can reduce the term in question to

$$x_1^2\,\varepsilon^{-1} + x_2^2\,\varepsilon^{-1} + \varepsilon^{-1} + 1 \tag{2.29}$$

Suppose $\varepsilon = 10^{-\alpha_o}$, $\alpha_o > o$; $x_1 = k_1(10)^{\alpha_1}$, and $x_2 = k_2(10)^{\alpha_2}$ such that $1 \leq k_i^2 < 10$, $i = 1, 2$. The restriction $1 \leq k_i^2 < 10$ merely prevents any change in the values of α_1 and α_2. Assume that $|\alpha_1| > |\alpha_2| > \alpha_o$, then the dominating term is $x_1^2\,\varepsilon^{-1}$ or

$$[k_1(10)^{\alpha_1}]^2[10]^{\alpha_o} \tag{2.30}$$

There are 56 bits of "magnitude information" in the IBM 360 system in double precision, ($2^{56} \approx 10^{17}$). To allow for overflow, use 10^{16}. In order to "prevent" round-off error the criterion to be used is

$$2\alpha_1 + \alpha_o < 16 \tag{2.31}$$

Since ε^{-1} appears in (2.29), it can be considered in (2.31)- this makes the calculations a little easier. Hence, equation (2.31) becomes

$$2\alpha_1 + 2\alpha_o < 16$$

or

$$\alpha_o < 8-\alpha_1 \tag{2.32}$$

The inequality (2.32) is only a guideline for choosing an initial ε.

This discussion can be generalized quite easily. Suppose that

$$\mu_{jk} = a_1 x_1^{\beta} + a_2 x_2^{\beta} + \ldots a_i x_i^{\beta} + \ldots + a_m \tag{2.33}$$

where $\beta > 0$ and $i = 1, 2, \ldots, d$. There will exist some x_i^{β} (or possibly some combination, say, $x_j^{\beta_1} x_i^{\beta_2} x_{i+2}^{\beta_3}$, where $\beta = \beta_1 + \beta_2 + \beta_3$), which may be written as,

$$x_i^{\beta} \triangleq [k_i (10)^{\alpha_i}]^{\beta} \tag{2.34}$$

such that α_i is greater than any other α_j. Following a similar procedure, one would determine the new guideline to be

$$2\beta\alpha_i + 2\alpha_o < 16 \tag{2.35}$$

2.6 Summary

To conclude this chapter, the sequential scheme will be outlined.

(1) Choose the form of the discriminant function.

(2) Choose a value for ε using (2.35).

(3) Choose the cost function, C(j/k).

(4) Apply equations (2.16, 2.17, 2.18, 2.19,) to the training set.

(5) After N samples, use (2.20) to determine the set of A_N^{jk} vectors.

(6) Determine the μ_{jk}'s from (2.13).

(7) Determine the M_N^{jk}'s from (2.21).

(8) Perform the decision process using (2.14).

CHAPTER III

A SEQUENTIAL ALGORITHM

3.1 Introduction

The program-called SAPRA, meaning Sequential Adaptive Pattern Recognition Algorithm,is divided into two parts: (1) Determining the A vectors and performance criterion, MN, and (2) The decision process. The former will be presented as a general flow table in this chapter and will be discussed in detail in Appendix A. The latter will be treated in detail in this chapter with the flow graph appearing, in detail, in Appendix B.

3.2 Generating the A Vectors

A general flow chart for the generation of the A vectors appears in Fig. 3.1. The following list presents each variable used and its associated meaning:

ε . . . . . convergence factor for P_o^{-1}

M . . . . . total number of terms in the discriminant function

NN. . . . . total number of training samples, NN=N

T . . . . . total number of categories , T=t

UPDATE. . . a "flag" used to update the system

COST. . . . the values of C(j/k)

MN_{jk}. . . . the performance criterion

D . . . . . the dimension of the space

The flow chart follows the following sequence of operations.

(1) The values of ε, M, NN, T, UPDATE, D, and COST are read in.

(2) The initial conditions for P_o^{-1} and θ_o^{jk} are set.

(3) Begin a "DO LOOP"; read one pattern vector and its associated category, n = 1, 2, . . ., NN.

(4) Compute $\phi_j(X_n)$, j = 1, 2, . . ., M.

(5) Compute P_n^{-1} and θ_n^{jk} and return to (3) above if n<NN.

(6) If n = NN, compute A_N^{jk}.

(7) Begin a "DO LOOP"; read one pattern vector and its associated category.

(8) Compute $\phi_j(X_n)$, j = 1, 2, . . ., M.

(9) Compute the value of the discriminant function μ_{jk}.

(10) Compute the "value" of the performance criterion, MN_{jk}, and return to (7) above if n<NN.

(11) If n = NN, compute <u>final</u> value of MN_{jk}.

3.3 The Decision Process

After the adaptive procedure is completed, i.e., the A vectors are determined, the decision rule is applied. Recalling, the rule states that for any X_i, i = 1, 2, . . ., N,

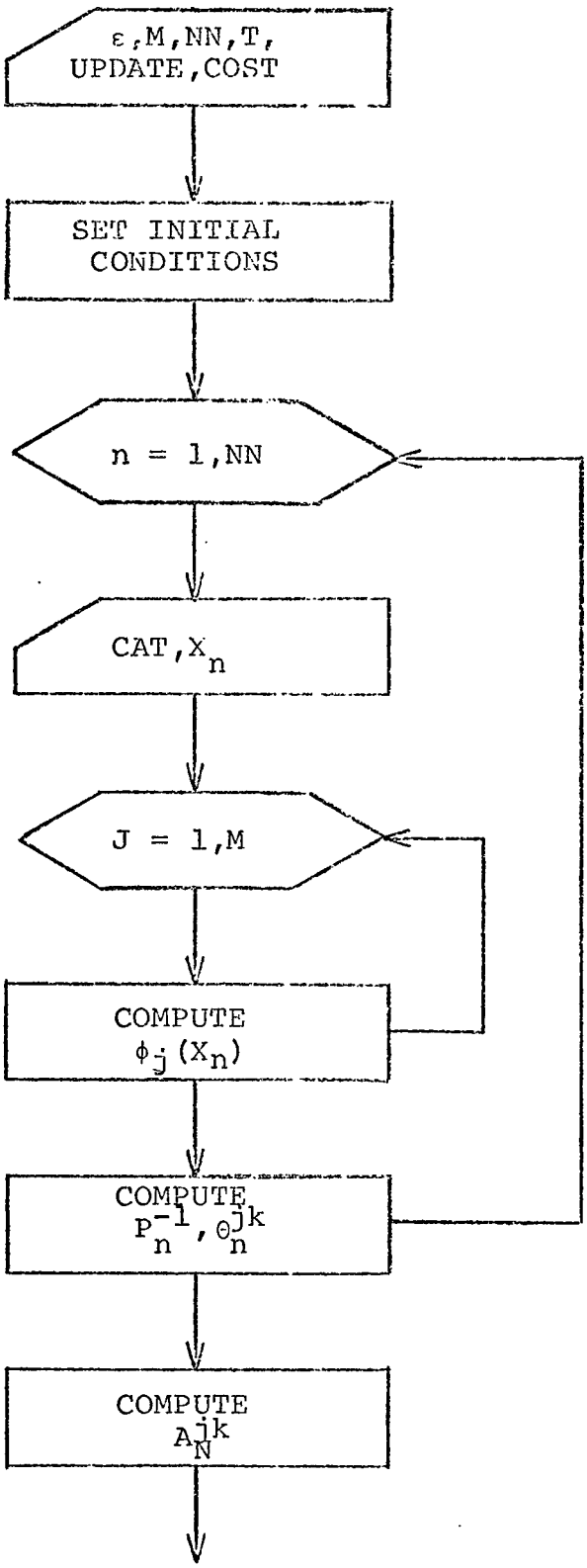


FIG. 3-1a

GENERAL FLOW CHART FOR SAPRA PROGRAM

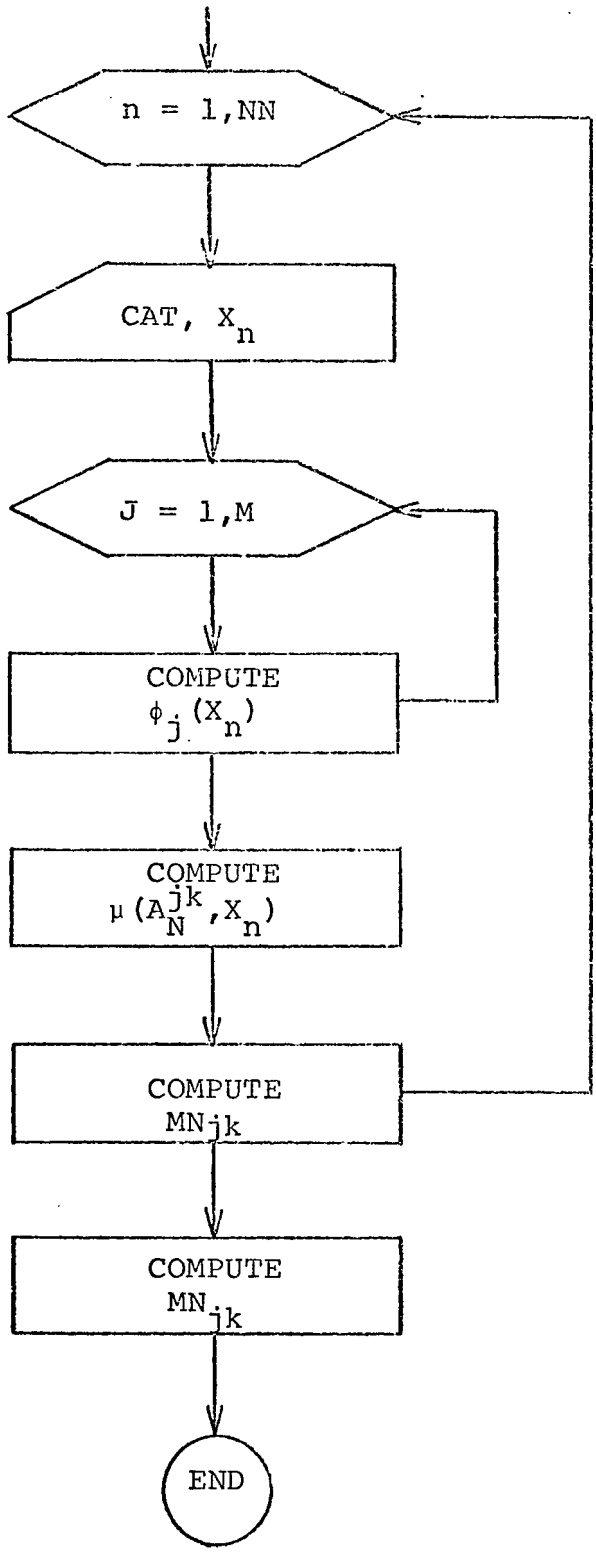


FIG. 3-1b

GENERAL FLOW CHART FOR SAPRA PROGRAM

if $\mu_{jk} \geq o$, then $X_i \in t_j$, otherwise $X_i \in t_k$, where $1 \leq j < k \leq T$. The bounds on "j" and "k" as applied to the discriminant functions mean that given a t-category problem, there will exist a set of discriminant functions as shown below, where $\mu_{jk} = \mu_{jk}(A_N^{jk}, X_i)$,

$$\mu_{12}\ \mu_{13}\ \mu_{14}\ \mu_{15} \cdots \mu_{1t}$$

$$\mu_{23}\ \mu_{24}\ \mu_{25} \cdots \mu_{2t}$$

$$\mu_{34}\ \mu_{35} \cdots \mu_{3t}$$

$$\cdots$$

$$\mu_{t-1,t}$$

Now, refer to row one (1) as the row containing all μ_{jk} where $j = 1$, row two (2) as the row containing all μ_{jk} where $j = 2$, etc; moreover, refer to column 2 as the column containing all μ_{jk} where $k = 2$, column 3 as the column containing all μ_{jk} where $k = 3$, etc. Assume that all μ_{jk} values have been determined and that $\mu_{1k_1} \geq o$ and $\mu_{1k_2} < o$, where $k_1 > 1, k_2 > 1$ and $k_1 \neq k_2$. Now, $\mu_{1k_1} \geq o$ implies that $X_i \in t_1$, whereas $\mu_{1k_2} < o$ implies that $X_i \in t_{k_2}$. This is considered to be a "conflict" since X_i can not belong to both t_1 and t_{k_2}. With this in mind, one can easily determine an algorithm to perform the decision making, or classification. The algorithm is as follows:

(1) If each $\mu_{1k} \geq o$, $k = 2, 3, \ldots, t$, then $X_i \in t_1$; otherwise, go to (2) below.

(2) If each $\mu_{2k} \geq o$, $k = 3, 4, \ldots, t$, then $X_i \in t_2$ iff $\mu_{12} < o$; otherwise go to (3) below.

(3) If there exists a row j' such that each $\mu_{j'k} \geq o$ then every $\mu_{jk} < o$, for $k = j'$, $j = 1, 2, \ldots j'-1$, is necessary and sufficient for $X_i \in t_{j'}$; otherwise go to (4) below.

(4) If (3) above can not be satisfied, then check the last column, i.e., if each $\mu_{jt} < o$, for $j = 1, 2, \ldots, t-1$, then $X_i \in t_t$; otherwise, go to (5) below. SEE NOTE AFTER (5)

(5) Since (4) above can not be satisfied, search for the smallest μ_{jk}, magnitude wise, and change its sign. Return to (1) above. If a classification does not result after the first sign change, then search for the "next" smallest value and change its sign. Return to (1) above. Continue this sign changing process until either a classification has been made or until all μ_{jk}'s have had their sign changed exactly one time.

NOTE: The value of μ_{jk} can be considered as a distance between X_i and the decision surface determined by μ_{jk}; hence, if one changes the sign of μ_{jk}, this has the effect of "moving" X_i from, say, t_j to t_k. If (4) above can not be satisfied, then a classification could possibly result by changing

the sign of the smallest μ_{jk}, magnitude wise. It should be noted that by changing the signs of the discriminant functions, one is "altering" the system; moreover, it seems that as the number of sign changes increases, the resulting classification, if any, will become somewhat questionable.

3.4 A Simulated Problem

A test problem was used to check the program's performance. A four-category, two dimensional problem was used for two reasons. (1) The multicategory aspect of the algorithm could be verified without using too much computer time. (2) A plot of each discriminant function could be obtained quite readily thus enabling a pictorial representation of the discriminatory process. The pattern vectors were chosen in a "random" manner; however, they were chosen so as to provide some overlapping between the categories. The cost function was chosen to be $C(j/k) = 1$, $j \neq k$ and $C(j/k) = 0.0$, $j = k$. Figures 3-2 and 3-3 display the simulated problem and the discriminant functions found using the SAPRA program.

3.5 Summary

In conclusion, this chapter has presented the basic flow chart for the SAPRA program; examined, in detail, the decision process, and presented the results of a simulated study involding a four-category two-dimensional space.

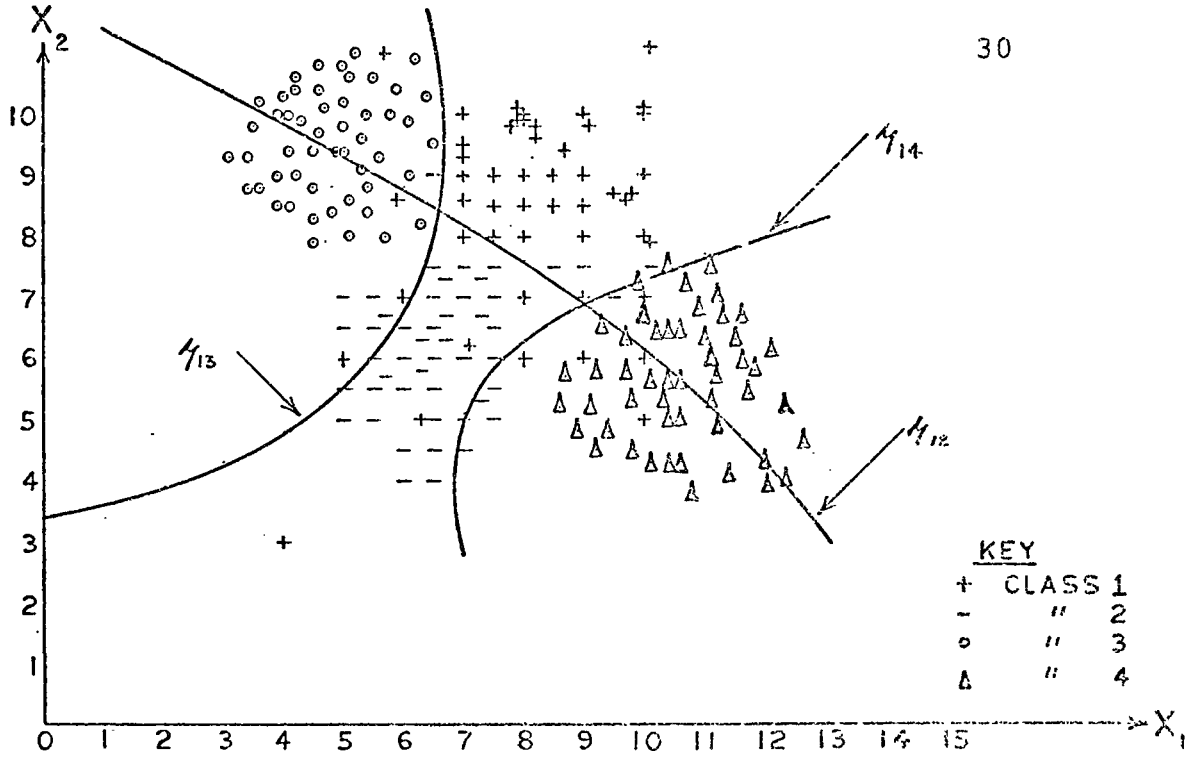


FIG. 3-3

PLOT OF DISCRIMINANT FUNCTIONS FOR SIMULATED PROBLEM

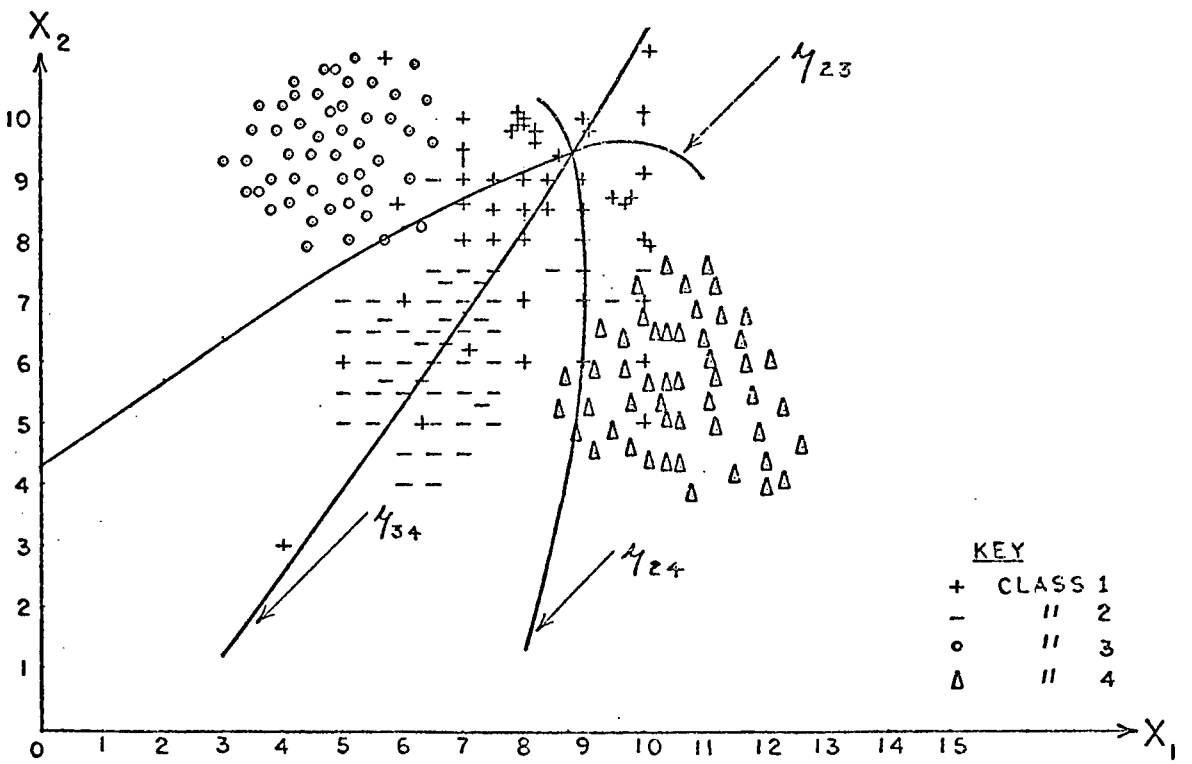


FIG. 3-2

PLOT OF DISCRIMINANT FUNCTIONS FOR SIMULATED PROBLEM

CHAPTER IV

DIFFERENTIATING REEF FROM NON-REEF CARBONATES

4.1 Introduction

It was felt that the results obtained from the simulated study of Chapter III indicated that the SAPRA program was performing satisfactorily. Hence, the next few chapters, IV thru VII, will be directed towards classification problems involving geophysical data.

4.2 The Problem

Chester [Ref. 1] was concerned with the problem of trying to determine the value of various trace elements as facies indicators in a series of reef and non-reef carbonates by analyzing the distribution of these elements. He obtained data from carbonates of Devonian Age from the Sturgeon Lake reef area of Alberta, Canada. In his analysis, he was interested in the detrital and non-detrital trace elements of the carbonates used. The word "detrital" implies that a rock mixture was carried into the basin of deposition in the solid state, whereas, non-detrital implies material entering the basin of deposition in solution. While removing the trace elements in the non-detrital portion of the rocks, one should be able to determine something about the chemical environment in which the sediment was formed. On the other

hand, the detrital fraction should reveal not only the nature of the source material but also the sedimentary characters which are more dependent on the physical variables of the area. Any trace element may be of some use as a geochemical indicator if (1) there is a significant variation in the content of the element as it is traced over a basin of deposition, (2) there is a relationship between the element and a certain type of environment.

4.3 The Method

The method used by Chester to determine the usefulness of trace elements as facies indicators is as follows:

(1) A Student's t-test was performed on the total number of analyses for each particular element from the two categories-reef, non-reef. The purpose of this test was to determine whether a statistical difference could be made between the two categories.

(2) A graph was made of the distribution of each element between the two populations.

(3) The trace element concentrations were partitioned by inspection according to the best separation.

(4) The accuracy of the chosen partition values was then evaluated.

4.4 The Results

The results of Chester's analysis on non-detrital trace elements will now be summarized. The trace elements, and element ratios, of concern were: Ni; Co; Cr; V; Cu; Pb; Ga; Ni/Co; Cr/V; Pv/Ga, and Cu/Co. There were 43 reef samples and 100 non-reef samples. The percentage of samples that were correctly classified from the reef facies was 71%, from the non-reef facies, 87%, and from both facies was 82%. The classification was performed using a combined 8-element, and element ratio, test or a 4-element, and element ratio, test for the samples in which Pb, Ga, and Cu were not determined. A scheme for classification was devised which assumed that a sample was correctly classified if five or more of the trace elements had values which fell within the limits for the facies of that sample. The 4-element test required that three values be within the bounds of the facies.

4.5 The Sequential Approach

When the SAPRA program was applied to this problem, the resulting classification was far superior to that obtained by Chester. The trace elements used were: Ni; Co; V; Cu, and Pb. Only 19 samples were used from the reef category and 60 samples, from the non-reef category. A quadratic discriminant function was chosen; hence, there were 21 terms in the function, i.e., $\phi_i(X)$ for $i = 1, 2, \ldots, 21$. The

initial choice for ε was determined using the boundary condition as established in Chapter II, 2.35. For convenience, the inequality (2.35) appears below.

$$2\beta\alpha_i + 2\alpha_o < 16 \qquad (2.30)$$

Since a quadratic discriminant function was chosen, $\beta = 2$. The value of α_i was determined from the data and found to be 2. Inserting these values into (2.35) yields $\alpha_o < 4$. A value of α_o which produced an acceptable performance criterion was $\alpha_o = 5$. The cost function was $C(j/k) = 1$, $j = k$ and $C(j/k) = 0$, whenever $j = k$. The performance criterion MN had a value of 0.258. The table below 4.1 presents a summary of the results obtained by the SAPRA program.

CATEGORY	NO. OF SAMPLES	NO. OF ERRORS
Reef	19	1
Non-Reef	60	2

TABLE 4.1

Differentiating Reef and Non-Reef Carbonates Using the SAPRA Program

The next table 4.2 compares the percentages of correct classification between Chester's method and SAPRA's method.

METHOD	CATEGORY	NO. OF SAMPLES	CORRECT CLASSIFICATION
Chester	Reef	43	71%
SAPRA	Reef	19	94%
Chester	Non-Reef	100	87%
SAPRA	Non-Reef	60	97%
Chester	Both	143	82%
SAPRA	Both	79	96%

TABLE 4.2

A Comparison Between Chester's Method and SAPRA's

It is clear from the above that the results from the SAPRA program are far superior to those obtained by Chester. Note, especially, that the results using SAPRA are better in all cases using fewer samples.

4.6 Summary

A problem concerned with differentiating between reef from non-reef carbonates using the analysis of non-detrital trace elements was presented. The percentage of correct classification obtained by using the SAPRA program was far superior to the method used by Chester. These results indicate that a geochemical differentiation can definitely be made between reef carbonates from non-reef carbonates.

CHAPTER V

SEPARATION OF SHALLOW AND DEEP EARTHQUAKES

5.1 The Problem

The problem is to differentiate between shallow (focus < 100 km deep) and deep (focus > 100 km deep) earthquakes. This was investigated by Nersesov et al in 1968 [Ref. 3].

5.2 The Method

He approached the problem in two ways.

(1) A separating function was formed by using the product of one-dimensional probability densities. This function was of the form

$$F(x_1, \ldots, x_n) = \frac{f_1(x_1)\ldots f_n(x_n)}{f_1(x_1)\ldots f_n(x_n)+g_1(x_1)\ldots g_n(x_n)} \tag{5.1}$$

The function (5.1) is used with seismic parameters, say, $z_1, \ldots, z_n$, such that if

$$F(z_1, \ldots, z_n) > K \tag{5.2}$$

where K is some constant such that $o \leq K \leq 1$, then the phenomenon is classified as belonging to Class I; and if

$$F(z_1, \ldots, z_n) \leq K \tag{5.3}$$

then it belongs to Class II.

(2) The second approach used a linear criterion of the following form:

$$F(x_1, \ldots, x_n) = \sum_{i,j=1}^{n} p_{ij} x_i (\alpha_i - A_j) \qquad (5.4)$$

where

n = the dimension of the space

p_{ij} = the inverse of the correlation matrix

α_i = the mean values of the parameters for Class I

A_j = the mean values of the parameters for Class II

The decision rule is similar to that used in (1) above with the exception that $-\infty < K < \infty$.

5.3 The Results

Nersesov et al used 100 training samples, 50 from each population, to adapt the system. The data was obtained from shallow and deep earthquakes in Central Asia. The parameters, dimensions, involved were (1) t_{max} - the time from first arrival to onset of the maximum in the "P" wave. (The "P" wave is the compressional wave; it travels in the direction of particle motion.) (2) f_p - the apparent frequency of the "P" wave. (3) f_s - the apparent frequency of the "S" wave. (The "S" wave, or shear wave, travels in a direc-

tion that is perpendicular to particle motion.)

After the system was adapted, Nersesov et al used 40 samples (20 from each class) to evaluate the probability of errors. The results were summarized and indicated that any one method produced approximately 10-15% erroneous classifications.

5.4 The Sequential Approach

The data which was used to verify Nersesov's system was used as the training set for the SAPRA program. Returning, once again, to the inequality (2.35), it was determined from the data used that $\alpha_i = 1$. Since a quadratic discriminant function was used, $\beta = 2$; and the resulting bound is $\alpha_o < 6$. The value of $\alpha_o = 5$ was chosen. The cost function was $C(j/k) = 1$ for $j \neq k$ and $C(j/k) = 0$ for $j = k$. Thus, with $\varepsilon = 10^{-5}$ and using 40 training samples, the value of the performance criterion was 0.305. Category I, deep earthquakes, had no errors in classification; Category II had 2 errors. Hence, a 5% error resulted.

By an analysis of the A vectors, one can determine the significance of any one, or more, of the parameters involved. For example, consider the following discriminant function

$$\mu = a_1x_1^2+a_2x_2^2+a_3x_1x_2+a_4x_1+a_5x_2+a_6 \qquad (5.5)$$

If, say, the values of a_1, a_3 and a_4 were such that a_1, a_3, $a_4 << a_2$, a_5, a_6, and if $x_1 << x_2$, then one would naturally assume that the terms $a_1x_1^2$, $a_3x_1x_2$ and a_4x_1 would contribute little to the discriminant function. Since the coefficients, a_1, . . ., a_6, are considered to be weighting factors, it is reasonable to assume that the parameter x_1 is of "lesser" significance than is x_2 for discriminatory purposes.

A situation of this nature occurred when applying the SAPRA program to this problem. The "insignificant" parameter was $x_1 = t_{max}$. Consequently, a second run with $\varepsilon = 10^{-5}$ was made using the same training set, but with $x_1 = o$. The results were gratifying; the performance criterion was 0.404. There were two errors in each category; thus, a total error of 10% resulted. This result strongly suggests an additional advantage of the SAPRA program, i.e., the capacity to differentiate between parameters that will "weigh heavily" in the discriminant function and those which will not.

5.5 Summary

A problem concerned with differentiating between deep and shallow earthquakes was investigated. The methods used by Nersesov et al produced an error of 10-15%. The SAPRA program not only produced fewer errors, 5%, but also used less

training samples for the adaptive process. The feasibility of using the A vectors to indicate "insignificant" training parameters was also suggested.

CHAPTER VI

TRACE ELEMENTS IN MARINE AND FRESH WATER

6.1 The Problem

The trace elements B, Co, Cr, Cu, Ga, Ni, Pb, V and Zn were analyzed in a total of 66 samples of both ancient and modern marine and fresh-water argillaceous sediments by Potter et al in 1962 [Ref. 7]. The purpose of the analysis was to determine the value, if any, of the above trace elements as environmental discriminators. The samples were taken such that the climates, sample location, and geologic age varied considerably in order to form a more general test. The data was categorized as follows: 14 samples of modern marine; 19 modern fresh; 20 ancient marine, and 13 ancient fresh water samples.

6.2 The Method

A preliminary graphical evaluation was first made by Potter et al for each of the trace elements. This involved choosing a partition line which would best separate modern marine samples from modern fresh water samples according to the concentration, in parts per million, of the trace elements. By using all six elements equally weighted, Potter et al were able to correctly classify 88% of the ancient sediments.

A statistical approach was taken next. A linear discriminant function of the form

$$X_7 = a_1x_1+a_2x_2+a_3x_3+a_4x_4+a_5x_5+a_6x_6+a_7x_7 \qquad (6.1)$$

was used, where x_1 = boron, x_2 = chromium, x_3 = copper, x_4 = gallium, x_5 = lead, x_6 = nickel, and x_7 = vanadium and the coefficients a_i, i = 1, . . ., 7, are the weighting factors. It was assumed that for each of the four categories there existed a common covariance matrix but that the means of the elements could vary. The common covariance matrix was obtained by a pooling of the covariance matrices for each of the four populations. The method of determining the weighting factors, a_i, for i = 1, . . ., 7, is discussed in Appendix I of Ref. 7. The basic approach was to determine "that linear combination of the x's for which the 't' of Student's test is maximized".

The classification scheme consisted of the following:

(1) Compute

$$X_{17} = a_1x_{11}+a_2x_{12}+ \ldots + a_7x_{17} \qquad (6.2)$$

$$X_{27} = a_1x_{21}+a_2x_{22}+ \ldots + a_7x_{27}$$

(2) Let $x_1, x_2, \ldots, x_7$ be observed for the unknown sample, then compute

$$X_7 = a_1x_1+a_2x_2+ \ldots + a_7x_7 \qquad (6.3)$$

(3) If X_7 has a value closer to that of X_{17}, then that sample belongs to population 1; and if X_7 has a value closer to that of X_{27}, then that sample is said to belong to category 2.

6.3 The Results

The results obtained by use of a linear discriminant function which used only the trace elements B and V were considered impressive. The form of the function was

$$X_2 = a_1x_1+a_2x_2 \qquad (6.4)$$

There were only 5 samples from the 33 "ancient" samples that were misclassified and 5 samples from the 33 "modern" samples that were misclassified.

6.4 The Sequential Approach

When the SAPRA program was applied to this problem, the results were even more impressive than were those obtained by Potter et al. Several different computer runs were made; the results of each will be discussed separately. The values for the cost function were $C(j/k) = 1$, $j \neq k$ and $C(j/k) = 0$, $j = k$.

RUN I

The elements used were B, Cr, Ni, and V. The elements Cr and V were analyzed by two different methods; hence instead of having a 4-dimensional problem, the problem had 6 dimensions. The value of $\alpha_o = 3$ was chosen. The categories were:

I = modern marine water

II = modern fresh water

III = ancient marine water

IV = ancient fresh water

A quadratic discriminant function was chosen. The resulting values of the performance criterion were, for $\varepsilon = 10^{-3}$:

$MN_{12} = 0.1864$

$MN_{13} = 0.1506$

$MN_{14} = 0.1678$

$MN_{23} = 0.1871$

$MN_{24} = 0.2963$

$MN_{34} = 0.1244$

The table below 6.1 indicates the errors that were made in classification.

CATEGORY	TOTAL ERRORS	ERROR CATEGORIES
I	2	II
II	3	IV
III	1	I
IV	5	I/1, II/4

TABLE 6.1

Classification Errors for the Elements B, Cr, Ni and V

Notice that 4 samples were classified as belonging to II when they actually belonged to category IV, and that 3 were put in IV when they belonged to II. This discrimination problem is indicated by the value of MN_{24} in comparison to the remaining MN values. The percentage of error is 16.7%.

RUN II

The elements used were B, Cr, Cu, Ga, Ni, and V. The discriminant function was quadratic, and $\varepsilon = 10^{-3}$. The categories were the same as those of Run I. The resulting values of the performance criterion were:

$MN_{12} = 0.2289$

$MN_{13} = 0.2190$

$MN_{14} = 0.1529$

$MN_{23} = 0.2052$

$MN_{24} = 0.2208$

$MN_{34} = 0.1652$

Table 6.2 below displays the errors made during classification. The error is 16.7%.

CATEGORY	TOTAL ERRORS	ERROR CATEGORIES
I	1	IV
II	2	IV
III	5	I/2, II/3
IV	3	I/1, II/2

TABLE 6.2

Classification Errors for the Elements

B, Cr, Cu, Ga, Ni, and V

RUN III

The elements used were B and V. The problem was subdivided as indicated below.

CATEGORY	CASE I	CASE II
I	ancient marine	modern marine
II	ancient fresh	modern fresh

Case I

The discriminant function was again quadratic, and $\varepsilon = 10^{-3}$. The value of $MN_{12} = 0.3736$ was determined. Table 6.3 indicates the errors made during classification and compares them to those made by Potter et al.

CATEGORY	SAPRA ERRORS	POTTER ERRORS
I	1	2
II	2	3

TABLE 6.3

Classification Errors for the Elements B and V

I = Ancient Marine; II = Ancient Fresh

The error percentage for SAPRA was 9% and for Potter et al, 15%.

Case II

The discriminant function was quadratic, and $\varepsilon = 10^{-3}$. The value of $MN_{12} = 0.3356$ was determined. Table 6.4 compares the errors made during classification.

CATEGORY	SAPRA ERRORS	POTTER ERRORS
I	1	1
II	1	4

TABLE 6.4

Classification Errors for the Elements B and V

I = Modern Marine; II = Modern Fresh

The percentage of errors are 6% for SAPRA and 15% for Potter et al.

Run IV

The elements used were B and V. The categories were

the same as those of run I. For $\varepsilon = 10^{-3}$, the values of the performance criterion were:

$$MN_{12} = 0.3417$$
$$MN_{13} = 0.3705$$
$$MN_{14} = 0.2831$$
$$MN_{23} = 0.3423$$
$$MN_{24} = 0.4564$$
$$MN_{34} = 0.2781$$

Table 6.5, below, presents the classification error made.

CATEGORY	TOTAL ERRORS	ERROR CATEGORIES
I	4	II/2, III/2
II	4	III/2, IV/2
III	7	I/4, II/3
IV	10	I/1, II/9

TABLE 6.5

Classification Errors for the Elements B and V

It is obvious that the results of this run are not acceptable. The conclusion is that the trace elements B and V are not good differentiators for the four-category case.

Run V

The elements B and V were used once again; however, the discriminant function was chosen to be linear. Further-

more, the problem was subdivided in the same manner as in Run III. The discriminant function had the form

$$\mu_{12} = a_1x_1+a_2x_2+a_3 \qquad (6.5)$$

where x_1 = boron and x_2 = vanadium. Table 6.6 presents the results of case I, where $\varepsilon = 10^{-4}$. The performance criterion was $MN_{12} = 0.4340$

CATEGORY	SAPRA ERRORS	POTTER ERRORS
I	3	2
II	2	3

TABLE 6.6

Classification Errors for the Elements B and V

I = Ancient Marine; II = Ancient Fresh

Table 6.7 shows the results for Case II, where $\varepsilon = 10^{-4}$ and $MN_{12} = 0.4135$.

CATEGORY	SAPRA ERRORS	POTTER ERRORS
I	2	1
II	2	4

TABLE 6.7

Classification Errors for the Elements B and V

I = Modern Marine; II = Modern Fresh

6.5 Summary

The question as to whether trace elements can be used as indicators to differentiate between both modern and ancient marine and fresh water has been studied. The results of several studies which employed the SAPRA program indicate quite strongly that certain trace elements may indeed be used as "water differentiators". The last table, 6.8, of this chapter consolidates all of the previous results concerned with this classification problem.

RUN	ELEMENTS	NUM. OF CATEGORIES	DISCRIMINANT FUNCTION	VALUE OF ε	% "SAPRA" ERRORS	% POTTER ERRORS
I	B,Cr,Ni,V	4	Quadratic	10^{-3}	16.7%	Not Applicable
II	B,Cr,Cu, Ga,Ni,V	4	Quadratic	10^{-3}	16.7%	Not Applicable
III	B,V	2 (Ancient)	Quadratic	10^{-3}	9%	15%
	B,V	2 (Modern)	Quadratic	10^{-3}	6%	15%
IV	B,V	4	Quadratic	10^{-3}	38%	Not Applicable
V	B,V	2 (Ancient)	Linear	10^{-4}	15%	15%
	B,V	2 (Modern)	Linear	10^{-4}	12%	15%

TABLE 6.8

Summary of Results for Chapter 6

CHAPTER VII

SAND ANALYSIS

7.1 The Problem

The last problem is concerned with the differentiation between modern beach, coastal dune, inland dune and river sands on the basis of a whole phi sieve analysis of these sands. In 1967, Moiola and Weiser [Ref. 2] made an evaluation on the use of textural parameters as indicators to classify the aforementioned grades of sand. A total of 120 samples, 30 from each of the categories, were gathered for the study. A sieve analysis was run on the data. Following this analysis, Moiola and Weiser compiled the quarter, half, and whole phi weight percentage distributions and then calculated the associated textural parameters by using a linear interpolation on the weight percentage distributions. The parameters used were:

(1) mean diameter - the average grain size

(2) standard deviation - a measure of dispersion about the mean

(3) skewness - a measure of asymmetry of distribution

(4) kurtosis - a measure of peakedness of the frequency curve

7.2 The Method

Moiola and Weiser then made plots of the textural

parameters using various combinations of the four categories. If a given plot demonstrated an effective differentiation between the two populations of concern, then a straight line, called a boundary line, was drawn such that a maximum separation between the two categories was attained. The results of their work will be presented with those attained by the use of the SAPRA program.

7.3 The Sequential Approach

The program was employed in three different ways on this problem. The first method considered various two-category problems as did Moiola and Weiser. The second approach investigated the idea of "stripping" one category from the four populations, one category from the remaining three, etc. The last method treated the entire problem as one four-category problem. The results of these approaches will be presented shortly.

The data that was used was obtained from a whole phi sieve analysis of 120 samples, 30 samples from each of the four categories. An analysis of the data reveiled that $\alpha_i = -4$. A quadratic discriminant function was used, hence, $\beta = 2$. And, from (2.35), $\alpha_o < 16$. The value $\alpha_o = 10$ was used for each of the three approaches to the problem. The cost function was $C(j/k) = 1$, $j \neq k$ and $C(j/k) = 0$, $j = k$.

Tables 7.1 thru 7.4 present the results of the various two-category combinations. The columns labled "whole", "half", and "quarter" refer to the errors of Moiola and Weiser for whole phi, half phi, and quarter phi sieve analysis data.

CATEGORY	SAPRA	WHOLE	HALF	QUARTER
Inland	0	5	3	2
Coastal	3	0	0	1

TABLE 7.1

Classification Errors for Two-Category Sand Problem

$MN_{12} = 0.1737$

CATEGORY	SAPRA	WHOLE	HALF	QUARTER
River	0	1	1	0
Beach	0	2	0	1

TABLE 7.2

Classification Errors for Two-Category Sand Problem

$MN_{12} = 0.0437$

CATEGORY	SAPRA	WHOLE	HALF	QUARTER
Beach	2	1	0	0
Inland	1	2	3	2

TABLE 7.3

Classification Errors for Two-Category Sand Problem

$MN_{12} = 0.1334$

CATEGORY	SAPRA	WHOLE	HALF	QUARTER
River	0	3	1	1
Coastal	0	0	0	0

TABLE 7.4

Classification Errors for Two-Category Sand Problem

$$MN_{12} = 0.0328$$

The results in table 7.1 indicate that three samples (SAPRA column) from the Coastal Category were misclassified. The error is 5%. Table 7.2 shows that there were no errors with the SAPRA program. Table 7.3 indicates that there were two beach samples classified incorrectly and one inland dune sample classified incorrectly. The error is 5%. Table 7.4 shows a perfect separation between coastal and river sands.

The next set of results were obtained by a "stripping" process. The concept of stripping can best be explained by the use of an example. Consider a three-category, say, A, B, and C, classification problem. One means of classification would be to separate (strip), say, class A from the remaining two categories, and then strip B from C. In other words, the stripping technique essentially re-defines the problem from a t-category problem to several two-category ones. In the example above, for the "first strip", Category

A could be considered as Category I and Categories B and C combined as Category II. For the "second strip", ategory B could be I and C, II. The order of stripping is important in that different errors result for different orders of stripping. The results of the stripping runs, tables 7.5 and 7.6, demonstrate this effect.

FIRST STRIP		SECOND STRIP		THIRD STRIP	
CATEGORY	ERRORS	CATEGORY	ERRORS	CATEGORY	ERRORS
I-River	2	I-Beach	7	I-Inland	0
II-Beach	4	II-Inland	9	II-Coastal	3
Inland		Coastal			
Coastal					

TABLE 7.5

Classification Errors Obtained by "Stripping"

FIRST STRIP		SECOND STRIP		THIRD STRIP	
CATEGORY	ERRORS	CATEGORY	ERRORS	CATEGORY	ERRORS
I-Beach	7	I-River	2	I-Inland	0
II-River	9	II-Inland	3	II-Coastal	3
Inland		Coastal			
Coastal					

TABLE 7.6

Classification Errors Obtained by "Stripping"

Note that the total number of errors for table 7.5 is 25 and for table 7.6 is 24. Note, also, that when beach

sands were stripped, in both cases more errors were attained than at any other strip.

The last approach viewed the situation as a four-category problem. The resulting values of the performance criterion were:

$$MN_{12} = 0.1454$$
$$MN_{13} = 0.1391$$
$$MN_{14} = 0.1568$$
$$MN_{23} = 0.1606$$
$$MN_{24} = 0.3946$$
$$MN_{34} = 0.2029$$

Table 7.7 displays the errors in classification.

CATEGORY	TOTAL ERRORS	ERROR CATEGORIES
I-River	4	III/3, IV/1
II-Beach	9	IV/9
III-Inland	5	I/2, II/1, III/2
IV-Coastal	8	II/8

TABLE 7.7

Classification Errors for Four-Category Sand Problem

Notice that 9 samples were classified as belonging to category IV when they actually belonged to II, and that 8 samples were misclassified in Category IV. This result is

not surprising since the value of MN_{24} is much larger with respect to the other values of the performance criterion. Also, note that more errors were again encountered with the classification of beach sands. Compare the errors of tables 7.5, 7.6, and 7.7. This consistency of misclassification for beach sand may indicate that the parameters used during the adaption process were very similar to those of coastal sand.

7.4 Summary

The last example problem has been concerned with the classification of modern beach, coastal dune, inland dune, and river sands on the basis of a whole phi sieve analysis of the samples used. Three different approaches were taken on this problem:

(1) considering various combinations of two-category problems;

(2) a "stripping" process;

(3) a four-category problem.

Moiola and Weiser approached the problem by method one above. They used the sieve analysis data to calculate the associated texture parameters, and then they performed the separation. The SAPRA program used the raw sieve analysis

data to perform the classification, thus bypassing any intermediate calculations. The results obtained with SAPRA had fewer misclassifications than did those of Moiola and Weiser.

CHAPTER VIII

CONCLUDING REMARKS

An adaptive multicategory pattern recognition algorithm was programmed using an IBM 360/44 digital computer. The algorithm was divided into two main programs: (1) The adaption process; (2) The decision process. The effectiveness of the algorithm was tested by using geophysical data gathered from various sources. In all, four problems were run with this "real-world" data; and in each case, the resulting classifications obtained were considered excellent.

The SAPRA program has a number of advantages over many of the presently used pattern classification schemes.

(1) The algorithm inverts the matrix of concern by a sequential routine. Thus less computer time and storage are required.

(2) A priori knowledge of the probability density functions or distributions is not necessary to perform a classification decision.

(3) There exists a number, or set of numbers if there are more than two categories, called the performance criterion which gives a strong indication as to how "good" the resulting classification will be.

(4) By a proper interpretation of the values of the co-

efficients in the discriminant function, one can determine the relative effectiveness of each training parameter.

(5) There exists the capability to "update" the values of the coefficients of the discriminant function without having to employ the data which has been used previously for training. The "update" procedure will be discussed in Appendix A.

(6) The number of categories and dimensions allowed in any given problem is only limited by economic considerations and the size of the computer being used.

(7) The classification becomes optimal as the number of training samples becomes large.

Several observations were made.

(1) An incorrect choice for ε can result in poor classification due to computer round-off error.

(2) If there are more than two categories, the values of the performance criterion relative to one another give an indication as to the difficulty encountered when classifying pattern vectors.

(3) As the order of the discriminant function increases, the resulting error of classification decreases.

BIBLIOGRAPHY

1. Chester, R.; Geochemical Criteria for Differentiating Reef from Non-Reef Facies in Carbonate Rocks; Bulletin of the American Association of Petroleum Geologists, Vol. 49, No. 3, March 1965, pp. 258-276.

2. Moiola, R. J. and Weiser, D.; Textural Parameters: An Evaluation; Journal of Sedimentary Petrology, Vol. 38, No. 1, March 1968, pp. 45-53.

3. Nersesov, I. L.; Pisarenko, V. F.; Rautian, T. G.; Smirnova, N. A., and Khalturin, V. I.; Studies on the Tsunami Wave, Joint Publications Research Service, Washington, D. C., 57P, 2 Mat 69 UNCLAS FLD/GP 8C.

4. Nilsson, N. J.; Learning Machines, McGraw-Hill Book Co., New York, 1965.

5. Patterson, J. D.; Wagner, T. S., and Womack, B. F., A Performance Criterion for Adaptive Pattern Classification Systems, Joint Automatic Control Conference Preprint, August 1966, pp. 38-46.

6. Pitt, J. M. and Womack, B. F.; Proceedings of the IEEE, December 1966, pp. 1987-1988.

7. Potter, P. E.; Shrimp, N. F., and Witters, J., Trace Elements in Marine and Fresh-Water Argillaceous Sediments, Geochimica et Cosmochimica Acta, 1963, Vol. 27, Pergamon Press Ltd., pp. 669-694.

8. Ralston, A., A First Course in Numerical Analysis, McGraw-Hill Book Co., New York, N. Y., 1965, p. 462, problem 65.

APPENDIX A

ADAPTIVE PROGRAM

A.1 A detailed analysis of the adaptive portion of the program will be presented in this appendix. The analysis will follow the program as listed at the end of this appendix.

This program was written to handle a problem with the following specifications:

(1) maximum number of categories = 4

(2) maximum number of dimensions = 6

(3) maximum order of the discriminant function = 2.

If one desires to solve a more complex problem, then the dimension statements may need to be modified. For example, consider a problem where there are 5 categories (T = 5), 7 dimensions (D = 7), and a second order discriminant function is chosen. Let the dimensions be represented by $x_1, \ldots, x_7$. The discriminant function will be

$$\begin{aligned}\mu_{jk} = {} & a_1x_1^2+a_2x_2^2+a_3x_3^2+a_4x_4^2+a_5x_5^2+a_6x_6^2+a_7x_7^2+a_8x_1x_2+a_9x_1x_3 + \\ & a_{10}x_1x_4+a_{11}x_1x_5+a_{12}x_1x_6+a_{13}x_1x_7+a_{14}x_2x_3+a_{15}x_2x_4 + \\ & a_{16}x_2x_5+a_{17}x_2x_6+a_{18}x_2x_7+a_{19}x_3x_4+a_{20}x_3x_5+a_{21}x_3x_6 + \\ & a_{22}x_3x_7+a_{23}x_4x_5+a_{24}x_4x_6+a_{25}x_4x_7+a_{26}x_5x_6+a_{27}x_5x_7+a_{28}x_6x_7 + \\ & a_{29}x_1+a_{30}x_2+a_{31}x_3+a_{32}x_4+a_{33}x_5+a_{34}x_6+a_{35}x_7+a_{36} \qquad \text{(A.1.1)}\end{aligned}$$

There are 36 terms in the discriminant function; therefore, M = 36. The dimensioning statements can now be modified according to T = 5, D = 7, and M = 36. For example, the A vectors would be dimensioned, as is suggested by the first group of comment cards, A(T-1, T, M) = A(4, 5, 36). Of course the user could dimension for the maximum limits that he would expect to handle in any given problem, but this wastes computer memory.

A.2 The variable names will now be defined.

M = the number of $\phi_i(X)$ functions appearing in the discriminant function

D = the dimension of the space, i.e., the number of elements in one pattern vector

T = the total number of categories

E = the convergence factor ε, $P_o^{-1} = \frac{1}{\varepsilon} I$

NN = N = the total number of training samples

CF(M) = the M different $\phi_i(X)$ functions appearing in the discriminant function, e.g., using equation (A.1.1),

$$CF(1) = \phi_1(X) = x_1^2$$

$$CF(2) = \phi_2(X) = x_2^2$$

$$\vdots$$

$$CF(34) = \phi_{34}(X) = x_6$$

$$CF(35) = \phi_{35}(X) = x_7$$

$$CF(36) = \phi_{36}(X) = 1$$

X(D) = a D-dimensional pattern vector

PNI(M,M) = the (mxm) matrix which is computed by equation (2.16)

TH(T-1,T,M) = the θ_n^{jk} vector(s) given by equation (2.9) for $1 \leq j < k \leq T$

A(T-1,T,M) = the A_N^{jk} vector(s) given by equation (2.20) for $1 \leq j < k \leq T$

The M, in each case designates the element of the vector. For example, suppose T = 4 and M = 28, then one would expect the following vector output, where

$$A(j,k,\ell) = A_N^{jk}$$

$$TH(j,k,\ell) = \theta_n^{jk}$$

for j = 1, . . ., T-1, k = j+1, . . ., T, ℓ=1, . . ., M, and $1 \leq j < k \leq T$

A(1,2,1)	=	TH(1,2,1)	=
A(1,2,2)	=	TH(1,2,2)	=
⋮		⋮	
A(1,2,28)	=	TH(1,2,28)	=
A(1,3,1)	=	TH(1,3,1)	=
A(1,3,2)	=	TH(1,3,2)	=
⋮		⋮	
A(1,3,28)	=	TH(1,3,28)	=
⋮		⋮	

A(1,4,28)	=	TH(1,4,28)	=
A(2,3,1)	=	TH(2,3,1)	=
A(2,3,2)	=	TH(2,3,2)	=
⋮		⋮	
A(3,4,28)	=	TH(3,4,28)	=

Z(T-1,T) = $Z^{jk}(n)$ as defined by equations (2.17) and (2.18) where $1 \leq j < k \leq T$.

COST(T,T) = a(T x T) matrix whose elements are the cost factors, C(j/k), which are determined by the user. Recall that C(j/k) = 0 whenever j = k for $1 \leq j < k \leq T$.

MN(T-1,T) = MN_{jk} = a(T-1xT) matrix whose elements are the performance criteria for $1 \leq j < k \leq T$.

MU(T-1,T) = μ_{jk} as defined by equation (2.13).

UPDATE = a "flag" used in the program to determine whether to read the initial values of P_o^{-1} and θ_o (UPDATE = 1) or to set the initial values of P_o^{-1} and θ_o equal to 0.0 (UPDATE = 0). This will be discussed in A.3.

Recall equation (2.16), which is repeated below for convenience.

$$P_n^{-1} = P_{n-1}^{-1} - (\Phi^T(n) P_{n-1}^{-1} \Phi(n) + 1)^{-1} P_{n-1}^{-1} \Phi(n) \Phi^T(n) P_{n-1}^{-1} \tag{2.16}$$

where n = 1, 2, . . ., N-1, N and $P_o^{-1} = \frac{1}{\varepsilon} I$. Now, isolate the term $(\Phi^T(n)P_{n-1}^{-1}\Phi(n)+1)^{-1}$. Define:

$$C1 = \Phi(n)^T P_{n-1}^{-1} = \text{a(1xm) vector}$$

$$C2 = C1\ \Phi(n) = \text{a(1x1) vector}$$

$$C = C2 + 1 = \text{a constant}$$

Now, isolate the term $P_{n-1}^{-1}\Phi(n)\Phi^T(n)P_{n-1}^{-1}$. Define

$$CFP = \Phi(n)\Phi^T(n) = \text{an (mxm) matrix}$$

$$P1 = CFP\ P_{n-1}^{-1} = \text{an (mxm) matrix}$$

$$P2 = P_{n-1}^{-1}P_1 = \text{an (mxm) matrix}$$

A.3 The operation of the program will now be discussed. Consider a problem where T = 4, D = 6, NN = 120 (30 samples from each of the four categories), and a quadratic discriminant function is chosen. Since D = 6, M = 28. The dimensioning statements would be the same as those in the program in the back of this appendix. Determine, from the data, the value of α_i and use inequality (2.35) to choose an α_o. For example, suppose that the largest piece of data was 145, and the smallest, 0.003. Then

$$145 = 1.45(10)^2, \quad \alpha_1 = 2$$

$$0.003 = 3.0(10)^{-3}, \quad \alpha_2 = -3$$

Since $|\alpha_2| > |\alpha_1|$, choose $\alpha_i = -3$. Since μ_{jk} is quadratic, $\beta = 2$. Thus, using (2.35)

$$2\beta\alpha_i + 2\alpha_o < 16$$

$$2(\partial)(-3) + 2\alpha_o < 16$$

$$-6 + \alpha_o < 8$$

$$\alpha_o < 14$$

Choose $\alpha_o = 10$; hence $\varepsilon = 10^{-\alpha_o} = 10^{-10}$. [REMEMBER: The "16" in (2.35) was determined from the fact that the IBM 360 had 56 magnitude bits ($2^{56} \simeq 10^{17}$). If a different computer is used, it may be necessary to re-define (2.35).] Now, determine the values of the cost functions, C(j/k), being sure that C(j/k) = 0.0 whenever $j = k$, $1 \leq j < k \leq T$.

Assume that this is the first time any of the training samples have been used for the adaptive procedure; then, UPDATE = 0. Refer to the program when necessary for the following sequence of operations.

(1) Dimension statements are read.

(2) Integer T, CAT, UPDATE, D.

(3) Read E, M, NN, T, UPDATE, D.

(4) Read cost functions.

(5) Set "upper bound" for subscript (superscript) "j".

(6) Initialize $\theta_o^{jk} = 0$.

(7) Initialize $P_o^{-1} = \frac{1}{\varepsilon} I$.

(8) Begin data loop, i.e., read one pattern vector.

(9) Calculate $\phi_i(X)$, i = 1, 2, . . ., 28.

(10) Begin calculation of P_n^{-1}.

(11) Determine C1.

(12) Determine C2.

(13) Determine C.

(14) Determine CFP.

(15) Determine P1.

(16) Determine P2.

(17) Determine P_n^{-1}.

(18) Determine $Z^{jk}(n)$.

(19) Determine $\theta^{jk}(n)$.

(20) If all pattern samples have been read, go to (21); otherwise, return to (8) above.

(21) Determine A_N^{jk} vectors.

(22) Set initial values of $MN_{jk} = 0.0$.

(23) Begin data loop, i.e., read one pattern vector.

(24) Determine $\phi_i(X)$, i = 1, 2, . . ., 28.

(25) Determine μ_{jk} values.

(26) Determine MN_{jk} values.

(27) Determine final MN_{jk} values.

(28) Punch an output deck containing the A_N^{jk} vectors, the θ_N^{jk} vectors and the matrix P_N^{-1}.

Now, suppose that additional training samples have become available, and you wish to update the system. The flag, UPDATE, is now set equal to one (1). The values that were obtained for the Θ_N^{jk} vectors and the P_N^{-1} matrix from the object deck will be the new initial conditions for the "update run". The program operation will be essentially the same, except in two operations.

(1) Steps 6 and 7 above will be modified, i.e., the initial values will be read from the object deck.

(2) Because the performance criteria are calculated by using the A_N^{jk} vectors plus all of the data required to calculate these vectors, the values of MN_{jk} calculated during an updating procedure are not valid. It is for this reason that a third program may be desired to calculate the MN_{jk} values, even though such a program is not necessary.

A.4 The order of the data cards for the adaptive program is given below.

UPDATE = 0

1. Card with E, M, NN, T, UPDATE, D. See format statement 500.

2. Cards with cost factors [C(j/k)]. See format statement 520. For example, if T = 4 and all cost functions are equal,

then

0.	1.	1.	1.	CARD 2
1.	0.	1.	1.	CARD 3
1.	1.	0.	1.	CARD 4
1.	1.	1.	0.	CARD 5

Note C(1/1) = C(2/2) = C(3/3) = C(4/4) = 0.

3. First set of data cards. See format statement 640.

4. Duplicate set of data. Same format.

UPDATE = 1

1. Card with E, M, NN, T, UPDATE, D. See format statement 500.

2. Cards with cost factors. See format statement 520.

3. Cards from output deck containing θ_N^{jk} vectors.

4. Cards from output deck containing P_N^{-1} matrix.

5. First set of data cards. See format statement 640.

6. Duplicate set of data. Same format.

```
C THE FOLLOWING PERTAINS TO THE DIMENSION STATEMENT:
C                          M=NUMBER OF PHI FUNCTIONS
C                          D=THE NUMBER OF DIMENSIONS IN THE SPACE
C                          T=THE NUMBER OF DISTINCT CATEGORIES
C                          E=THE CONVERGENCE FACTOR TO CALCULATE PNI
C                          CF(M)
C                          CFP(M,M)
C                          C1(M)
C                          TH(T-1,T,M)
C                          A(T-1,T,M)
C                          P2(M,M)
C                          PNI(M,M)
C                          P1(M,M)
C                          X(D)
C                          Z(T-1,T)
C                          COST(T,T)
C                          MN(T-1,T)
C                          MU(T-1,T)
C NN IS THE NUMBER OF PATTERN VECTORS OF KNOWN CLASSIFICATION
      DOUBLE PRECISION CF(28),CFP(28,28),C1(28),TH(3,4,28),A(3,4,28),
     1P1(28,28),P2(28,28),PNI(28,28),E,C2,C,MN(3,4),MU(3,4)
      DIMENSION X(6),Z(3,4),COST(4,4)
      INTEGER T,CAT,UPDATE,D
C     READING IN SYSTEM PARAMETERS
      READ(5,500)E,M,NN,T,UPDATE,D
  500 FORMAT(1D10.9,5I5)
      WRITE(6,510)E,M,NN,T,UPDATE,D
  510 FORMAT(1H1,25X,' E=',D20.9,5X,'M=',I5,5X,'NN=',I5,5X,'T=',I5,5X,'
     1UPDATE=',I5,5X,'D =',I2,/////)
C     READING IN COST CRITERIA
      DO 10 I=1,T
   10 READ(5,520) (COST(I,J),J=1,T)
  520 FORMAT(16F5.2)
      WRITE(6,530)
  530 FORMAT(50X,'THE COST MATRIX IS',//)
      DO 20 L1=1,T
   20 WRITE(6,540)(COST(L1,M1),M1=1,T)
  540 FORMAT(54X,16(1X,F5.2))
C     FIXING UPPER BOUND FOR SUBSCRIPT "J"
      JJ=T-1
      DO 30 J0=1,JJ
      JJ0=J0+1
      DO 30 K0=JJ0,T
      IF(UPDATE.EQ.1)GO TO 5
      DO 40 I0=1,M
   40 TH(J0,K0,I0)=0.
      GO TO 30
C     READING IN INITIAL THETA VECTOR
    5 READ(5,570)(TH(J0,K0,I0),I0=1,M)
  570 FORMAT(4D20.10)
   30 CONTINUE
      DO 60 J1=1,M
      IF(UPDATE.EQ.1)GO TO 6
      DO 70 K1=1,M
```

```
      PNI(J1,K1)=0.
   70 PNI(J1,J1)=1./E
      GO TO 60
C     READING IN INITIAL MATRIX
    6 READ(5,600)((PNI(J1,K1),K1=1,M)
  600 FORMAT(4D20.10)
   60 CONTINUE
C     BEGINNING DATA LOOP
      DO 90 N=1,NN
      READ(5,640) CAT,(X(J),J=1,D)
  640 FORMAT(I5,9F10.3)
C     CALCULATING PHI FUNCTION VALUES
      CF(1)=X(1)*X(1)
      CF(2)=X(2)*X(2)
      CF(3)=X(3)*X(3)
      CF(4)=X(4)*X(4)
      CF(5)=X(5)*X(5)
      CF(6)=X(6)*X(6)
      CF(7)=X(1)*X(2)
      CF(8)=X(1)*X(3)
      CF(9)=X(1)*X(4)
      CF(10)=X(1)*X(5)
      CF(11)=X(1)*X(6)
      CF(12)=X(2)*X(3)
      CF(13)=X(2)*X(4)
      CF(14)=X(2)*X(5)
      CF(15)=X(2)*X(6)
      CF(16)=X(3)*X(4)
      CF(17)=X(3)*X(5)
      CF(18)=X(3)*X(6)
      CF(19)=X(4)*X(5)
      CF(20)=X(4)*X(6)
      CF(21)=X(5)*X(6)
      CF(22)=X(1)
      CF(23)=X(2)
      CF(24)=X(3)
      CF(25)=X(4)
      CF(26)=X(5)
      CF(27)=X(6)
      CF(28)=1.
C     BEGINNING CALCULATION FOR INVERSE OF P MATRIX
      DO 100 J2=1,M
      C1(J2)=0.
      DO 100 K2=1,M
  100 C1(J2)=CF(K2)*PNI(K2,J2)+C1(J2)
      C2=0.
      DO 110 J3=1,M
  110 C2=C1(J3)*CF(J3)+C2
      C=C2+1.
      DO 120 J4=1,M
      DO 120 K4=1,M
  120 CFP(J4,K4)=CF(J4)*CF(K4)
      DO 130 J5=1,M
      DO 130 K5=1,M
```

```
      P1(J5,K5)=0.
      DO 130 L5=1,M
  130 P1(J5,K5)=CFP(J5,L5)*PNI(L5,K5)+P1(J5,K5)
      DO 140 J6=1,M
      DO 140 K6=1,M
      P2(J6,K6)=0.
      DO 140 L6=1,M
  140 P2(J6,K6)=PNI(J6,L6)*P1(L6,K6)/C+P2(J6,K6)
      DO 150 J7=1,M
      DO 150 K7=1,M
  150 PNI(J7,K7)=PNI(J7,K7)-P2(J7,K7)
C     END OF CALCULATIONS FOR INVERSE OF P MATRIX
C     DETERMINING THETA VECTOR
      DO 160 J8=1,JJ
      JJ1=J8+1
      DO 160 K8=JJ1,T
      IF(CAT.NE.J8.AND.CAT.NE.K8)Z(J8,K8)=COST(K8,CAT)-COST(J8,CAT)
      IF(CAT.EQ.J8)Z(J8,K8)=COST(K8,J8)
      IF(CAT.EQ.K8)Z(J8,K8)=-COST(J8,K8)
      DO 170 L8=1,M
  170 TH(J8,K8,L8)=TH(J8,K8,L8)+Z(J8,K8)*CF(L8)
  160 CONTINUE
   90 CONTINUE
      WRITE(6,660)
  660 FORMAT(//,50X,'THE A(J,K) AND TH(J,K) VECTORS ARE',///)
C     DETERMINING A VECTOR
      DO 180 J9=1,JJ
      JJ2= J9+1
      DO 180 K9=JJ2,T
      DO 180 L9=1,M
      A(J9,K9,L9)=0.
      DO 190 M9=1,M
  190 A(J9,K9,L9)=PNI(L9,M9)*TH(J9,K9,M9)+A(J9,K9,L9)
  180 WRITE (6,670) J9,K9,L9,A(J9,K9,L9),J9,K9,L9,TH(J9,K9,L9
  670 FORMAT(30X,'A(',I2,',',I2,',',I2,')',1X,'=',1X,D20.10,10.
     1'TH(',I2,',',I2,',',I2,')',1X,'=',1X,D20.10,/)
C     SETTING INITIAL VALUE OF PERFORMANCE CRITERION EQUAL TO ZERO
      DO 220 J=1,JJ
      KK=J+1
      DO 220 K=KK,T
  220 MN(J,K)=0.
C     BEGINNING DATA LOOP
      DO 240 I=1,NN
      READ(5,730) CAT,(X(J),J=1,D)
  730 FORMAT(I5,9F10.3)
C     CALCULATING PHI FUNCTIONS VALUES
      CF(1)=X(1)*X(1)
      CF(2)=X(2)*X(2)
      CF(3)=X(3)*X(3)
      CF(4)=X(4)*X(4)
      CF(5)=X(5)*X(5)
      CF(6)=X(6)*X(6)
      CF(7)=X(1)*X(2)
      CF(8)=X(1)*X(3)
```

```
      CF(9)=X(1)*X(4)
      CF(10)=X(1)*X(5)
      CF(11)=X(1)*X(6)
      CF(12)=X(2)*X(3)
      CF(13)=X(2)*X(4)
      CF(14)=X(2)*X(5)
      CF(15)=X(2)*X(6)
      CF(16)=X(3)*X(4)
      CF(17)=X(3)*X(5)
      CF(18)=X(3)*X(6)
      CF(19)=X(4)*X(5)
      CF(20)=X(4)*X(6)
      CF(21)=X(5)*X(6)
      CF(22)=X(1)
      CF(23)=X(2)
      CF(24)=X(3)
      CF(25)=X(4)
      CF(26)=X(5)
      CF(27)=X(6)
      CF(28)=1.
C     CALCULATING VALUE OF DISCRIMINANT FUNCTION
      DO 250 J=1,JJ
      KK=J+1
      DO 250 K=KK,T
      MU(J,K)=0.
      IF(CAT.NE.J .AND.CAT.NE.K )Z(J ,K )=COST(K ,CAT)-COST(J ,CA
      IF(CAT.EQ.J )Z(J ,K )=COST(K ,J )
      IF(CAT.EQ.K )Z(J ,K )=-COST(J ,K )
      DO 260 L=1,M
  260 MU(J,K)=A(J,K,L)*CF(L)+MU(J,K)
C     CALCULATING PARTIAL VALUE OF PERFORMANCE CRITERION
  250 MN(J,K)= (MU(J,K)-Z(J,K))**2+MN(J,K)
  240 CONTINUE
      WRITE(6,740)
  740 FORMAT(///,50X,'THE PERFORMANCE CRITERION IS:',///)
      DO 270 J=1,JJ
      KK=J+1
      DO 270 K=KK,T
C     CALCULATING FINAL VALUE OF PERFORMANCE CRITERION
      MN(J,K)=MN(J,K)/ NN
  270 WRITE(6,750)J,K,MN(J,K)
  750 FORMAT(50X,'MN(',I2,',',I2,') =',D20.10,///)
C     PUNCH OUTPUT DECK FOR A VECTORS,THETA VECTORS AND PNI MATRI>
      DO 280 J=1,JJ
      KK=J+1
      DO 280 K=KK,T
  280 WRITE(7,760)(A(J,K,L),L=1,M)
  760 FORMAT (4D20.10)
      DO 290 J=1,JJ
      KK=J+1
      DO 290 K=KK,T
  290 WRITE(7,760)(TH(J,K,L),L=1,M)
      DO 300 J=1,M
  300 WRITE(7,760)(PNI(J,K),K=1,M)
      WRITE(7,8888)
 8888 FORMAT(' ')
      END
```

APPENDIX B

THE DECISION PROCESS

B.1 Since the decision process has been discussed in detail in section 4.3, only the flow chart, Fig. B-1, the definitions of variables, the program, and the order of the data cards will appear in this appendix.

DATA CARDS

B.2

1. Card containing T, M, D. See format statement 200.
2. Cards containing A_N^{jk} vectors from output deck.
3. Data cards. See format statement 240.

PROGRAM VARIABLES

KOUNTS = maximum number of sign changes

JJ = upper bound for superscript/subscript "j"

KOUNT = counter for sign changes

LATCH = a controller used when finding "smallest" value of μ_{jk}

SMALL 1 = used with LATCH and SMALL 2 to determine "smallest" μ_{jk}

CAT = category

SMALL 2 = used with LATCH and SMALL 1 to determine "smallest" μ_{jk}

J STAR, K STAR = subscripts of "smallest" μ_{jk}

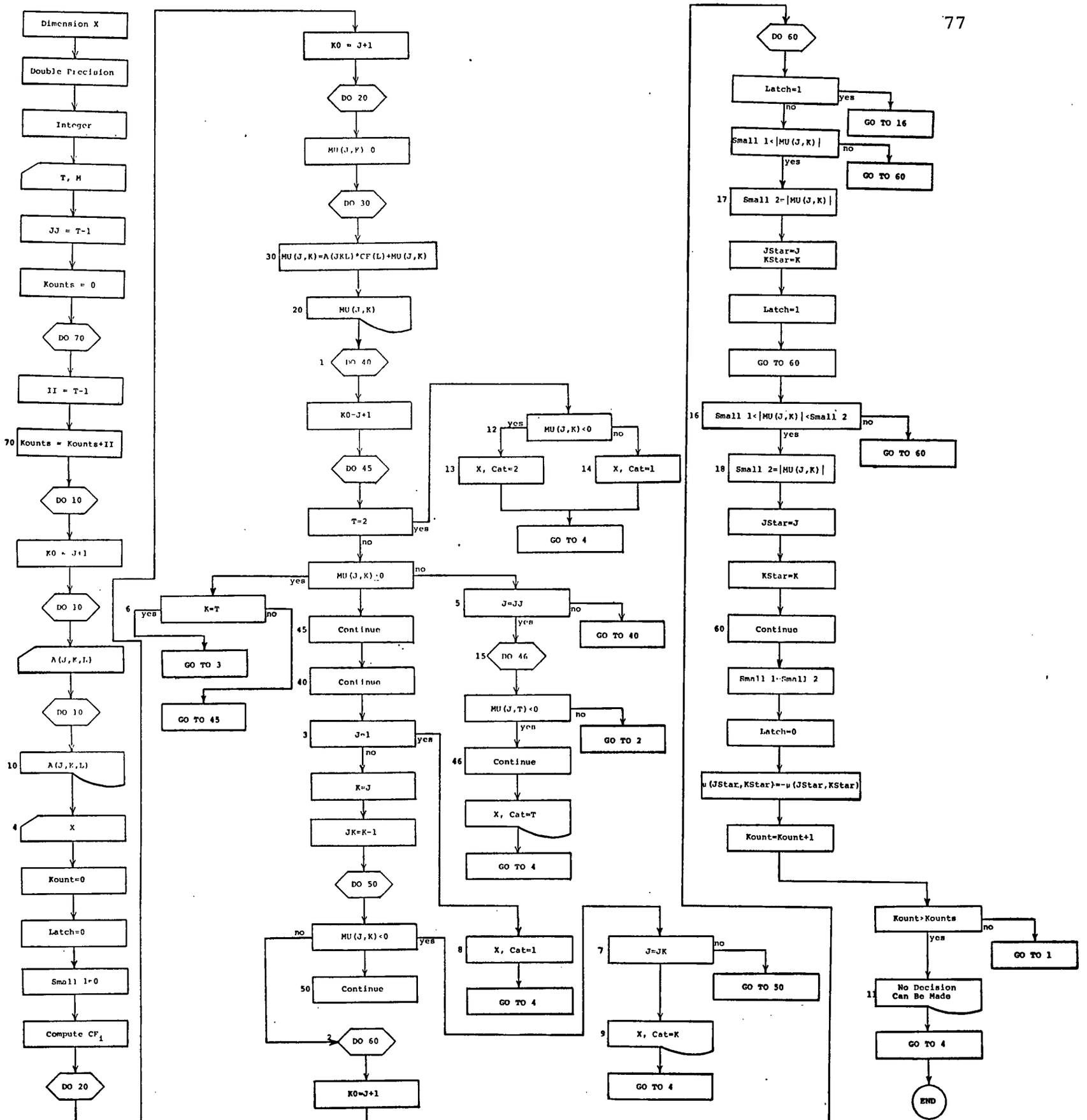


FIG. B-1 FLOW GRAPH FOR DECISION PROCESS

```
C  M = THE NUMBER OF PHI FUNCTIONS.
C  D = THE DIMENSION OF THE PATTERN SPACE.
C  THE FOLLOWING PERTAINS TO THE DIMENSION STATEMENT: CF(M),X(D),
C  MU(T-1,T),A(T-1,T,M),CF(M)
      DOUBLE PRECISION MU(3,4),CF(28),A(3,4,28),SMALL1,SMALL2
      DIMENSION X(6)
      INTEGER T,TWO/2/,ONE/1/,D
      READ(5,200) T,M,D
  200 FORMAT(3I5)
      WRITE(6,210)T,M,D
  210 FORMAT(1H1,50X,'T',1X,'=',I2,5X,'M',1X,'=',I2,5X,'D =',I2,//)
      JJ=T-1
      KOUNTS=0
      DO 70 I=1,T
      II=T-I
   70 KOUNTS=KOUNTS*II
      DO 10 J=1,JJ
      KO=J+1
      DO 10 K=KO,T
      READ(5,220)(A(J,K,L),L=1,M)
  220 FORMAT(4D20.10)
      DO 10 L=1,M
   10 WRITE(6,230)J,K,L,A(J,K,L)
  230 FORMAT(50X,'A(',I2,',',I2,',',I2,')',1X,'=',1X,D20.10,/)
    4 READ(5,240)(X(I),I=1,D)
  240 FORMAT(5X,6F10.5)
      KOUNT=0
      LATCH=0
      SMALL1=0.
      CF(1)=X(1)*X(1)
      CF(2)=X(2)*X(2)
      CF(3)=X(3)*X(3)
      CF(4)=X(4)*X(4)
      CF(5)=X(5)*X(5)
      CF(6)=X(6)*X(6)
      CF(7)=X(1)*X(2)
      CF(8)=X(1)*X(3)
      CF(9)=X(1)*X(4)
      CF(10)=X(1)*X(5)
      CF(11)=X(1)*X(6)
      CF(12)=X(2)*X(3)
      CF(13)=X(2)*X(4)
      CF(14)=X(2)*X(5)
      CF(15)=X(2)*X(6)
      CF(16)=X(3)*X(4)
      CF(17)=X(3)*X(5)
      CF(18)=X(3)*X(6)
      CF(19)=X(4)*X(5)
      CF(20)=X(4)*X(6)
      CF(21)=X(5)*X(6)
      CF(22)=X(1)
      CF(23)=X(2)
      CF(24)=X(3)
      CF(25)=X(4)
```

```
      CF(26)=X(5)
      CF(27)=X(6)
      CF(28)=1.
      DO 20 J=1,JJ
      KO=J+1
      DO 20 K=KO,T
      MU(J,K)=0.
      DO 30 L=1,M
   30 MU(J,K)=A(J,K,L)*CF(L)+MU(J,K)
   20 WRITE(6,255)J,K,MU(J,K)
  255 FORMAT(50X,'MU(',I2,',',I2,') =',D20.10)
    1 DO 40 J=1,JJ
      KO=J+1
      DO 45 K=KO,T
      IF(T.EQ.2) GO TO 12
      IF(MU(J,K))5,6,6
    5 IF(JJ-J)15,15,40
    6 IF(T-K)3,3,45
   45 CONTINUE
   40 CONTINUE
    3 IF(J.EQ.1) GO TO 8
      K=J
      JK=K-1
      DO 50 J=1,JK
      IF(MU(J,K))7,2,2
    7 IF(JK-J)9,9,50
    9 WRITE(6,270) (X(I),I=1,D)
  270 FORMAT(33X,6(F9.4,3X))
      WRITE(6,271)K
  271 FORMAT('+',100X,'CAT =',I2)
      GO TO 4
   50 CONTINUE
   15 DO 46 J=1,JJ
      IF(MU(J,T).GE.0.) GO TO 2
   46 CONTINUE
      WRITE(6,270)(X(I),I=1,D)
      WRITE(6,271)T
      GO TO 4
    2 DO 60 J=1,JJ
      KO=J+1
      DO 60 K=KO,T
      IF(LATCH.EQ.1)GO TO 16
      IF(SMALL1.LT.DABS(MU(J,K)))GO TO 17
      GO TO 60
   17 SMALL2=DABS(MU(J,K))
      LATCH=1
      JSTAR=J
      KSTAR=K
      GO TO 60
   16 IF(DABS(MU(J,K)).LT.SMALL2.AND.DABS(MU(J,K)).GT.SMALL1)GO TO 18
      GO TO 60
   18 SMALL2=DABS(MU(J,K))
      JSTAR=J
      KSTAR=K
```

```
 60 CONTINUE
    SMALL1=SMALL2
    LATCH=0
    MU(JSTAR,KSTAR)= MU(JSTAR,KSTAR)*(-1.)
    KOUNT=KOUNT+1
    IF(KOUNT.GT.KOUNTS) GO TO 11
    GO TO 1
 11 WRITE(6,280)
280 FORMAT(50X,'NO DECISION CAN BE MADE.')
    GO TO 4
 12 IF(MU(J,K))13,14,14
 13 WRITE(6,270)(X(I),I=1,D)
    WRITE(6,271)TWO
    GO TO 4
 14 WRITE(6,270)(X(I),I=1,D)
    WRITE(6,271)ONE
    GO TO 4
  8 WRITE(6,270) (X(I),I=1,D)
    WRITE(6,271)ONE
    GO TO 4
    END
```