NONPARAMETRIC TRAINING PROCEDURES FOR MULTICATEGORY PATTERN CLASSIFICATION

Thesis for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
ALBERT YEN-SHIEN HUNG
1971

This is to certify that the

thesis entitled

NONPARAMETRIC TRAINING PROCEDURES FOR
MULTICATEGORY PATTERN CLASSIFICATION

presented by

Albert Yen-Shien Hung

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Electrical
Engineering

Major professor

Date Sept. 30, 1971

set of

M reg

space

A set

crimin

fundam

the or

assume

unsupe

posed.

lating

a funct

This g

cases.

approxi

functio

to the

ABSTRACT

NONPARAMETRIC TRAINING PROCEDURES FOR
MULTICATEGORY PATTERN CLASSIFICATION

By

Albert Yen-shien Hung

The M-class pattern recognition problem is to construct a set of discriminant functions which partition a feature space into M regions, one region per pattern class. Each point in the feature space is a potential pattern and each pattern represents an object. A set of training patterns is to be generalized into a set of discriminant functions which classify the potential patterns. The fundamental algorithms developed here concern the situation where the origin of each training pattern is known and almost nothing is assumed about the origins of the patterns. An extension to the unsupervised case is also given.

Several new multi-class decision-making algorithms are proposed. An entirely new class of algorithms is obtained by translating the pattern recognition problem into the problem of minimizing a function of several variables and selecting suitable functions. This general formulation includes most known algorithms as special cases. The class of algorithms includes all procedures which approximate discriminant functions by linear combinations of basis functions. Several successful two-class algorithms are extended to the M-class problem.

The concept of linear inequalities and the role of the mean-square error criterion in pattern recognition are studied. Several algorithms are shown to rely heavily on the basic mean-square-error criterion. In order to solve the generalization problem, the conditional probabilities are selected to form the optimal discriminant functions. A class of multi-class algorithms using stochastic approximation techniques is proposed that learn the unknown coefficients of the discriminant functions. A digital simulation has been performed.

The most novel aspect of this thesis is the introduction of an algorithm that combines cluster-seeking and multiclass pattern recognition. Cluster-seeking tries to uncover the structure inherent in the training patterns. The algorithm exploits this structural information to construct discriminant functions. The success of the discriminant function in classifying training patterns then provides clues about structure. The algorithm is straightforward and computationally realistic. It has been tested with both artificial and practical data.

NONPARAMETRIC TRAINING PROCEDURES FOR MULTICATEGORY PATTERN CLASSIFICATION

By

Albert Yen-shien Hung

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering
and Systems Science

1971

to

have

to h

my e

Mich

Resea

resea

Dr. J

criti

the co

ACKNOWLEDGEMENTS

I am deeply indebted to Dr. R.C. Dubes who introduced me to the field of pattern recognition and whose valuable suggestions have gone a long way in shaping this thesis. I am also grateful to him for the guidance and encouragement I received from him during my entire doctoral degree program.

I am also indebted to the Division of Engineering Research, Michigan State University, and the Air Force Office of Scientific Research for their financial support during the course of this research.

Thanks are also due to Dr. R.O. Barr, Dr. G.L. Park, Dr. J.H. Stapleton and Dr. Rita Zemach for their interest and critical reviews of this thesis.

Finally, this thesis would be incomplete without mentioning the contribution of my wife Lilian to whom this thesis is dedicated.

C

II

IV

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

The problem of automatic data analysis has drawn considerable attention since the development of high speed computers. Today there are automated systems that read handwriting and fingerprints, as well as systems that classify data, forecast weather and perform medical diagnosis. In all these cases, some type of patterns or templates are assumed as a basis for recognition; that is, they are pattern recognition problems (H1, B3, N1, N2).

1.1 Basic Problems

There are three fundamental problems associated with pattern recognition.

(1) Feature extraction. A set of real variables $\{x_1,x_2,\ldots,x_d\}$, called features or attributes, identify the object to be classified. A vector $X^* = (x_1,x_2,\ldots,x_d)^T$ is called a pattern vector or, simply, a pattern. No general procedure currently exists for selecting an optimal set of features for a given problem. In most cases, the success of feature extraction depends entirely upon factors in the specific problem. The feature extraction problem will not be discussed in this thesis; a "good" set of features is assumed.

(2) The abstraction problem. Once a set of features has been selected, and a set of pattern vectors

$D^* = \{(X_i^*, y_i),\ i = 1,2,\ldots,N\}$ are given where y_i denotes the classification of the pattern vector X_i^*, then a set of augmented pattern vectors $D = \{(X_i, y_i),\ i = 1,2,\ldots,N\}$, called training patterns, can be formed. An augmented pattern vector is a d-dimensional vector X^* augmented by a (d+1)st component whose value is 1; that is, $X^T = (X^*, 1)^T$. The problem considered in this report is to find a set of discriminant functions $\{f_j(X)\}_{j=1}^{M}$ defined on the augmented feature space such that

$$f_i(X) > f_j(X) \quad \text{if} \quad X \in \text{class} \quad i;\ i,j = 1,2,\ldots,M,\ i \neq j. \qquad (1)$$

If there is a set of discriminant functions satisfying (1) then an "optimal" solution exists rather than one which permits a few violations of (1).

The abstraction problem is thus one of distilling the information from the training patterns that is necessary to construct a set of discriminant functions. Generally speaking, there are two approaches to the abstraction problem. First, if a great deal of information is available about the data, a probabilistic model can be generated to describe the underlying physical phenomenon. The abstraction problem can then be treated in the framework of statistical decision theory. This approach is the parametric (or statistical) method. Second, there are many problems, especially in the biomedical area, in which data has simply been amassed. It is reasonable to assume, however, that there exists a set of discriminant functions which approximate (1), and so the functional form for the discriminant functions is assumed known except for a set of parameters. Based on the given training

patt

lear

the

(or

prim

crim

to pr

was

probl

nonpa

is us

able

patte

mathe

1.2

genera

origin

litera

recogn

lem.

a coll

severa

on its

ultic

patterns, an iterative procedure, sometimes called training or learning, can be formulated to determine reasonable values for the unknown parameters. This approach is termed the nonparametric (or deterministic) method of training. This thesis will be primarily concerned with nonparametric training.

(3) The generalization problem. Once a set of discriminant functions has been determined, an attempt may be made to predict performance for new patterns. If the parametric method was chosen to solve the abstraction problem, the generalization problem consists in determining the error probability. If the nonparametric approach was adopted, the generalization question is usually difficult to answer because of the absence of a suitable criterion. The misclassification percentage with training patterns is usually employed even though it cannot be related mathematically to the problem parameters.

1.2 Multiclass Problems

A single data source, called a pattern class or category, generates each pattern. In the two-class problem, each pattern originates in one of two pattern classes. A review of the current literature (H2, H3, P1) indicates that the majority of pattern recognition algorithms have been developed for the two-class problem. In theory, the M-class $(M > 2)$ problem can be solved as a collection of $\binom{M}{2}$ two-class problems (N1). However, there are several advantages in considering M-class pattern classification on its own merits.

(1) A generalization of a two-class algorithm to solve a multiclass problem by pair-wise operations is quite involved

computationally. For instance, it will be necessary to compute $\binom{M}{2}$ generalized inverse matrices in the Ho-Kashyap algorithm (H2); with an M-class algorith, only one inverse computation is required.

(2) Two-class algorithms give no immediate answer to the multiclass problem even if a solution exists. A voting scheme must be imposed to classify a pattern. With an M-class algorithm, there is no need for a voting scheme and it will furnish a reasonable suboptimal solution in case of overlapping sets of training patterns.

(3) With a deterministic M-class algorithm, the problems of pattern recognition and cluster-seeking can be combined, as discussed later, into a single problem. Such a simplification is the main reason for devoting this thesis to deterministic M-class algorithms. Criteria and procedures are presented in Sec. 5.2.

1.3 Cluster-Seeking and Pattern Recognition

According to Ball (B1), "A cluster is a set of patterns contained in a high dimensional space where the density of patterns is large compared to the density in the surrounding volumes." The idea behind cluster-seeking techniques is the grouping of patterns into clusters or groups so that all patterns within a cluster are very "alike" and patterns in different clusters are very "unalike". Several measures of simularity will lead to clustering algorithms. For instance, Freidman and Rubin (F1) proposed some invariant criteria for grouping data through linear transformations. Ball and Hall (B2) suggested the distance

between a pattern and the cluster center be used. A number of other measures of simularity were categorized by Ball (B1). In this thesis, the criterion of minimum probability of misclassification will be used for cluster-seeking. A pattern recognition system is proposed which combines cluster analysis and classification in such a way that knowledge of data structure guides pattern classification, and the results of pattern classification provide additional insight into the true structure of the data.

Assuming that the feature extraction problem has been solved, a set of (augmented) training patterns $D = \{(X_i, y_i), i = 1,2,\ldots,N\}$ from M pattern classes is provided. The value of y_i, which indicates the true classification of X_i, may or may not be given for all i. The functional block diagram of the proposed system is shown in Figure 1.

If the classifications of all training patterns are known, the training patterns from each class are grouped to form one cluster per class. A preselected M-class iterative learning algorithm is applied to learn the unknown parameters in a set of M discriminant functions. The rate of learning is determined by properties of the algorithm and the distribution of patterns. If the performance of the algorithm is acceptable, a set of discriminant functions is obtained which adequately classifies all training patterns. By comparing the results of this classification with $\{y_i\}$, the misclassification percentage can be computed. If the performance is acceptable, the pattern recognition problem is solved and the data structure obtained assigns one cluster to each pattern class. Otherwise, all the misclassified patterns from

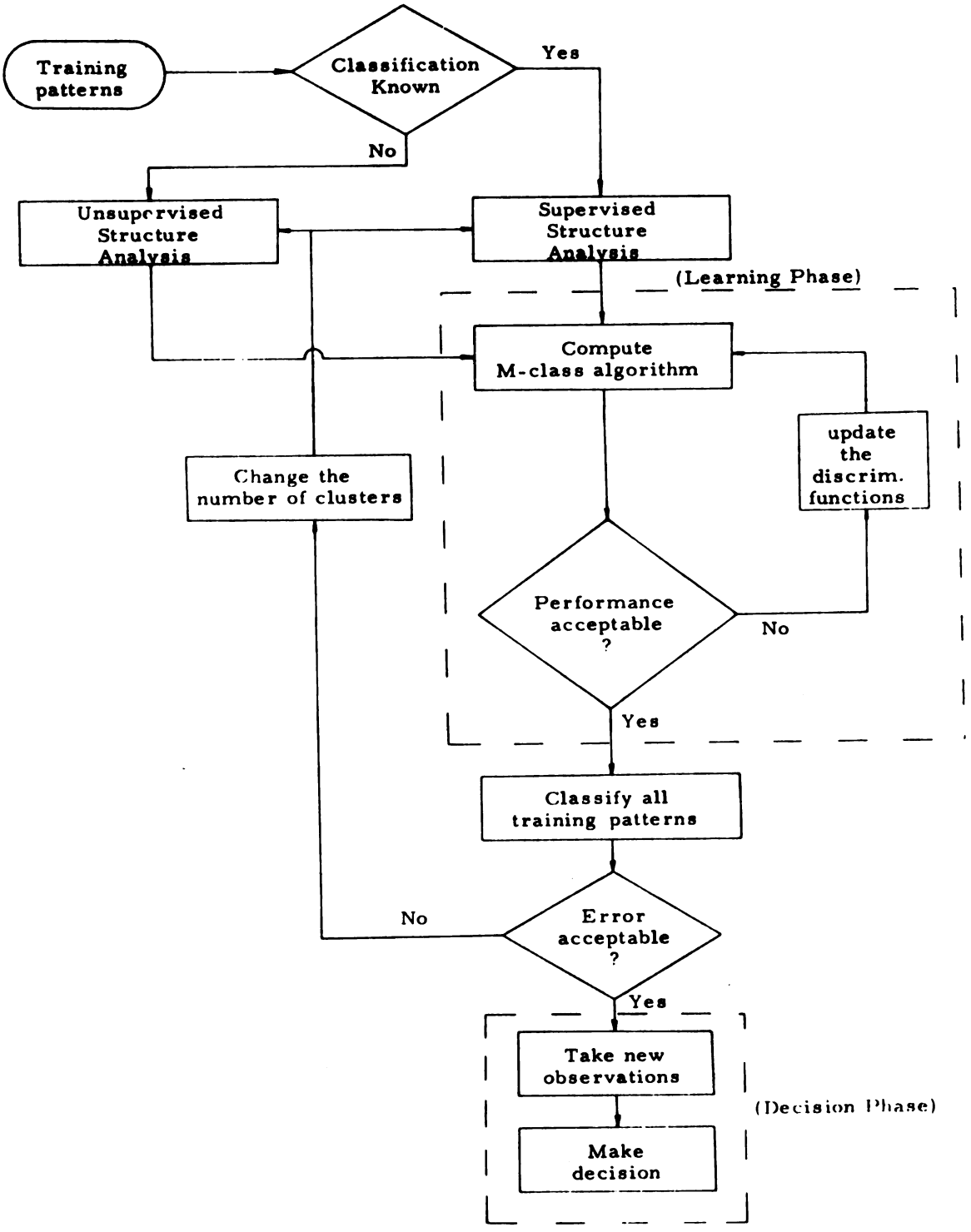


Figure 1. A proposed pattern recognition system.

each class are clustered. These new clusters together with the clusters of correctly classified patterns represent an updated data structure. A new set of discriminant functions is then obtained. The learning process will not stop until performance is satisfactory.

If the true pattern classifications are unknown, it is possible in principle but not in practice to solve the computational problem of evaluating all the possible partitions of input data into M clusters in order to find a partition that minimizes the probability of misclassification (F1). A suboptimal procedure for unsupervised structure analysis is discussed in Sec. 5.3. In any event, the system is switched from the learning phase to the classification phase when satisfactory system performance has been reached with training patterns.

1.4 Thesis Objectives and Outline.

This thesis suggests that cluster-seeking and multiclass pattern recognition can be combined into a workable pattern recognition system, Figure 1, in which the two problems are solved in a step-wise fashion, partial solutions for one providing clues for the other. The literature for both clustering and multiclass, nonparametric pattern recognition studies is reviewed, and a specific algorithm is proposed in Sec. 5.2. Several new multiclass algorithms are proposed as extensions of two-class algorithms.

In order to realize this proposed system, a class of linear multiclass learning algorithms is derived in Chapters II, III, and IV. The idea of translating the abstraction problem into an optimization problem in which linear functionals are minimized

under

creat

imagi

of li

form

respe

rithm

parti

patte

is sc

the g

as a s

ditio

as the

approx

unkno

tion a

sensit

abstra

tion s

cluste

artifi

under certain constraints provides a great deal of freedom when creating new algorithms. In fact, the only limitation is one's imagination in finding meaningful linear functionals. The concept of linear inequalities and the mean-square-error criterion for formulating linear functionals are studied in Chapters II and III, respectively. Based on these criteria, a general learning algorithm for the M-class problem is derived. In each chapter, particular algorithms are formulated as special cases.

Since no assumptions are made about the origins of the patterns in Chapters II and III, the pattern recognition problem is solved only for the given training patterns. In order to solve the generalization problem, we must view the fixed training patterns as a sample from a population as was done in Chapter IV. The conditional probability functions $P(\omega_i|X)$, $i = 1,2,\ldots,M$, are selected as the optimal discriminant functions. The techniques of stochastic approximation (W5) are employed to estimate the coefficients of unknown discriminant functions. Several M-class stochastic approximation algorithms are proposed. The relations among them and the sensitivity problems are investigated. Chapter V examines the abstraction problem in unstructured situations. A pattern recognition system which combines multiclass pattern recognition and cluster-seeking is proposed. The system has been tested with both artificial and practical data.

tic

$(W_i$

f_i

aug

If

det

mir

sir

tra

is

ine

tra

col

2.1

Pat

Con

CHAPTER II

GENERALIZED LINEAR INEQUALITIES IN MULTICATEGORY PATTERN RECOGNITION

The most important task in nonparametric pattern recognition can be posed as the selection of a set of weight vectors $\{W_i\}$ that defines a set of discriminant functions $\{f_i(X) = W_i^T \phi(X)\}$, where $\phi(X)$ is a vector function of the augmented pattern X; that is, the output of a "ϕ-machine" (N1). If training patterns are provided, the unknown weights can be determined either with or without training. For instance, a minimum-distance classifier (N1) is achieved without training since the unknown cluster points are computed directly from the training patterns non-recursively. In this section, training is viewed as an optimization problem and the concept of linear inequalities is chosen to form linear functionals. A general training algorithm for the M-class problem is derived and a collection of training algorithms are presented as special cases.

2.1 The Concept of Linear Inequalities in Pattern Recognition

In the two-class pattern recognition problem, a set of N patterns with known classification is given. The problem is to construct a discriminant function $f(X)$, such that

$$\begin{aligned} f(X) &> 0 \quad \text{if} \quad X \in \text{class} \quad \omega_1 \\ &< 0 \quad \text{if} \quad X \in \text{class} \quad \omega_2 \end{aligned} \tag{2}$$

f

(

th

ac

The

$\frac{1}{2}^T$ (

sel

and

func

the

patt

can

simu

Here

class

either

reul i

tions

approx

for as many of the N training patterns as possible.

By the Weierstrauss theorem on polynomial approximation (W1), one can choose a polynomial which uniformly approximates the continuous function $f(X)$ on a closed interval with arbitrary accuracy. The function $f(X)$ will be approximated by $\Phi(X)$

$$\Phi(X) = \sum_{i=1}^{d+1} C_i \Phi_i(X) = C^T \phi(X).$$

The parameters $C^T = (C_1, C_2, \ldots, C_{d+1})$ are unknown; $\Phi^T(X) = (\phi_1(X), \phi_2(X), \ldots, \phi_d(X), 1)$ are linearly independent, preselected, real functions. Nilsson (N1) calls $\Phi(X)$ a "Φ-function," and any pattern classifier employing Φ functions, a "Φ-machine."

It is not necessary to actually approximate the discriminant function $f(X)$ itself. It is sufficient to achieve agreement in the signs of the functions $f(X)$ and $\Phi(X)$ for all training patterns. Thus, the problem of finding a discriminant function can be converted to the problem of solving the system of N simultaneous linear inequalities in (3).

$$yC^T\phi(X) > 0 \quad \text{for all training patterns } X . \tag{3}$$

Here $y = 1$ if X is from class ω_1 and $= -1$ if X is from class ω_2. The coefficients in C^T are chosen to map $\phi(X)$ into either 1 or -1 as dictated by y. Once C^T is chosen, the decision reul is: Decide X is in ω_1 if $C^T\phi(X) > 0$, ω_2 if $C^T\phi(X) < 0$.

In M-class pattern recognition, a set of discriminant functions $\{f_i(X)\}_{i=1}^{M}$ is needed. Each discriminant function will be approximated by a finite sum

Once

λ_i

for

M

siona

to pro

natrua

a set

follo

Thus,

e_1 f

exists

X fro

$$\Phi_i(X) = \sum_{j=1}^{d+1} a_{ij}\phi_j(X) = a_i^T \phi(X) \quad ; \quad i = 1,2,\ldots,M \ .$$

Once $\{a_i^T\}$ is chosen, the decision rule is: Decide X is in ω_i if i is the smallest integer for which $a_i^T\phi(X) \geq a_j^T\phi(X)$ for all $j \neq i$.

The matrix of unknown parameters is written as the $M \times (d+1)$ matrix A.

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_M^T \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \cdots a_{1,d+1} \\ a_{21} & a_{22} \cdots a_{2,d+1} \\ \vdots \\ a_{M1} & a_{M2} \cdots a_{M,d+1} \end{bmatrix}$$

The letters $\{e_1,e_2,\ldots,e_M\}$ denote a set of M m-dimensional points, called vertices, for which $e_i^T e_j = 0$ for all $i \neq j$.

By analogy to the two-class problem, matrix A is chosen to project $\phi(X)$ onto the vertex e_i as dictated by Y_i, the natrual generalization of y; that is, $\{Y_i\}$ $(i = 1,2,\ldots,M)$ is a set of $M \times M$-dimensional orthogonal matrices which satisfy the following conditions.

$$\begin{aligned} Y_i^T Y_i &= I \\ Y_i e_i &= e_1 \end{aligned} \qquad \text{for all } i = 1,2,\ldots,M \ .$$

Thus, the orthogonal matrix Y_i transforms the vertex e_i into e_1 for each of the M pattern classes.

If the M pattern classes are linearly separable, there exists a linear transformation A which maps each training pattern X from class ω_i close to the vertex e_i. Choosing the vertex

$e_1 > 0$ (each entry of e_1 is greater than zero), an orthonormal matrix Y_i is to be found such that $Y_i A_\phi(X) \cong e_1$ for all X in class ω_i.

The natural extension of (3) to the M-class problem is:

$$Y_i A_\phi(X) > 0 \text{ (vector)} \quad \text{for all } i \text{ and } X \text{ when } X \text{ is in class } \omega_i \tag{4}$$

This is a system of MN simultaneous linear inequalities.

The inequalities (4) can be written in another form.

$$\begin{aligned} |C^T_\phi(X) - 1| &< |C^T_\phi(X) + 1| \quad \text{if } X \in \text{class } \omega_1 \\ |C^T_\phi(X) - 1| &> |C^T_\phi(X) + 1| \quad \text{if } X \in \text{class } \omega_2 . \end{aligned} \tag{5}$$

An extension of (5) to the M-class problem was proposed by Chaplin and Levadi (C1).

$$\|A_\phi(X) - e_i\| < \|A_\eth(X) - e_j\| \quad \text{if } X \in \text{class } \omega_i \text{ for all } j = 1,2,\ldots,M;\ i \neq j \tag{6}$$

where $\{e_i\}_{i=1}^M$ are M-dimensional vertices satisfying the following conditions:

$$\|e_i\| \equiv \text{trace}\{e_i\, e_i^T\} = 1$$

$$\|e_i - e_j\| = \|e_i - e_k\| \quad \text{for all } j,k \neq i .$$

Equation (6) can be rewritten as

$$e_i^T A_\phi(X) > e_j^T A_\eth(X) \quad \text{if } X \in \text{class } \omega_i \text{ for all } j \neq i$$

for all training pattern X. Equivalently, decide X is in class ω_i if

$$e_i^T A\phi(X) - \max_{j \neq i} \{e_j^T A\phi(X)\} > 0 \qquad (7)$$

Inequalities (3) and (4) are based on the idea of achieving agreement between the signs of discriminant functions and approximating functions. The inequality proposed by Chaplin and Levadi is based on a minimum-distance criterion. In both cases, the assumption of linear separability assures the existence of a linear transformation A which correctly classifies all N training patterns. When training patterns overlap, Equations (4) and (7) will be inconsistent, and no solution for a matrix A can be found.

2.2 The General Learning Algorithm

It is a common practice in solving linear inequalities to transform the problem into an optimization problem. Then, the gradient descent method, or some other optimization procedure, is chosen to minimize certain linear functionals. A general approach for obtaining meaningful linear functionals will now be described; these functionals will then be minimized to obtain general learning algorithms. The two-class pattern recognition problem will be considered first, and the results extended to the M-class problems.

Equation (3) can be rewritten as

$$yC^T\phi(X) = \beta > 0 \ ; \ \beta \text{ is a positive number .} \qquad (8)$$

Define

$$Z = yC^T\phi(X) - \beta \quad \text{for all} \quad X .$$

A strictly convex and differentiable function defined on the set Z is denoted by $F(.)$ where

$$F(Z) > 0 \quad \text{if} \quad Z < 0$$
$$= 0 \quad \text{if} \quad Z \geq 0 .$$

A solution to the equation $Z = 0$ can be obtained by minimizing a linear functional

$$J(C) = L\{F(Z)\} = L\{F(yC^T\phi(X) - \beta)\}$$

where L denotes a linear operator.

It is clear that the functional $J(C)$ will also be strictly convex and differentiable, which guarantees the existence and uniqueness of a minimum. The problem is now to find C such that $J(C)$ is a global minimum for the N training patterns.

The usual method for minimizing $J(C)$ is the gradient descent procedure (B3, D1). An algorithm can be written as

$$C[n] = C[n-1] - \rho \left\{ \frac{\partial J}{\partial C} \Big|_{C[n-1],X[n]} \right\} \tag{9}$$

where n is the iteration index, and $X[n]$ is the training pattern presented to the algorithm at the nth iteration; ρ is a positive scalar which determines the distance to be moved at each step in the direction of the negative gradient of $J(C)$ in the parameter space.

Equation (9) leads to the following general iterative algorithm for the two-class pattern recognition problem.

(GA.1)
$$C[n] = C[n-1] + \rho L\{F'(\epsilon_{n-1})y[n]\phi(X[n])\}$$
$$\beta[n] = \beta[n-1] + \rho L\{|F'(\epsilon_{n-1})| - F'(\epsilon_{n-1})\}$$

where
$$\epsilon_{n-1} = y[n]C^T[n-1]\phi(X[n]) - \beta[n-1]$$

for all $n = 1,2,3,\ldots$ and $\beta[0] > 0$.

It should be noted that, in the above algorithm, only one training pattern is used at each iteration. Each of the N training patterns must be presented to the algorithm many times to assure that J(C) is a global minimum.

Algorithm (GA.1) can be extended easily to the M-class problem by invoking (4). Equation (4) can be rewritten as

$$Y_i A_\phi(X) = \gamma > 0 \quad \text{for all } i, X .$$

Letting $W = Y_i A_\phi(X) - \gamma$, a solution of the linear equations $W = 0$ can be obtained by finding the minimum of the linear functional:

$$J(A) = L\{F(W)\} = L\{F(Y_i A_\phi(X) - \gamma)\} .$$

Again using (9), a general learning algorithm for solving the M-class pattern recognition problem is obtained.

(GA.2)
$$A[n] = A[n-1] + \rho L\{F'(\epsilon_{n-1}) Y_i^T[n] \phi(X[n])\}$$

$$\gamma[n] = \gamma[n-1] + \rho L\{|F'(\epsilon_{n-1})| - F(\epsilon_{n-1})\}$$

where
$$\epsilon_{n-1} = Y_i[n] A[n-1] \phi(X[n]) - \gamma[n-1] .$$

Euqation (GA.2) holds for all X in class ω_i, i = 1,2,...,M; and n = 1,2,...,; $\gamma[0] > 0$.

Similarly, (7) can be rewritten as

$$e_i^T A_\phi(X) - \max_{j \neq i} e_j^T A_\phi(X) = \beta > 0 . \tag{10}$$

This leads to the following general learning algorithm:

$$\text{(GA.3)} \qquad A[n] = A[n-1] + \rho L\{F'(\epsilon_{n-1})\}[e_i \phi^T(X[n]) - \max_{j \neq i} e_j \phi^T(X[n])\}$$

$$\beta[n] = \beta[n-1] + \rho L\{|F'(\epsilon_{n-1})| - F'(\epsilon_{n-1})\}$$

where $$\epsilon_{n-1} = e_i^T A[n-1]\phi(X[n]) - \max_{j \neq i} e_j^T A[n-1]\phi(X[n] - \beta[n-1]).$$

Equation (GA.3) holds for all X in class ω_i where $i = 1,2,\ldots,M$ and $n = 1,2,\ldots$.

In two-class pattern recognition, setting $e_1 = 1$, $e_2 = -1$, and $A = C^T$ changes (10) to:

$$2C^T\phi(X) = \beta \quad \text{if} \quad X \in \text{class } \omega_1$$

$$-2C^T\phi(X) = \beta \quad \text{if} \quad X \in \text{class } \omega_2 .$$

This is equivalent to:

$$yC^T\phi(X) = \beta/2, \qquad y = 1 \quad \text{if} \quad X \in \text{class } \omega_1$$

$$-1 \quad \text{if} \quad X \in \text{class } \omega_2 .$$

This is exactly (8). Therefore, (GA.3) reduces to (GA.1) for two-class problems.

Based on the general learning algorithms (GA.1), (GA.2), and (GA.3), a particular class of iterative algorithm can be derived easily. In fact, the number of specific algorithms is limited only by the availability of meaningful convex functions $F(.)$ and linear operators L. In Sec. 2.3, a class of standard iterative algorithm for the M-class problem is presented.

2.3 A Class of Iterative Procedures for Linear Inequalities

A class of iterative algorithms for the two-class problem corresponding to (GA.1) with $\beta[n] \equiv 0$ for all n is given in

Table 2 of Devyaterikov, Propoi, and Tsypkin (D1). The extension to M-class algorithms according to the general algorithm (GA.2) is straightforward and hence not included. A collection of iterative algorithms corresponding to (GA.3) will be presented in this section.

2.3.1 The Fixed Increment Method

Consider the following linear functional.

$$J(A) = \frac{1}{2}\{|e_i^T A\phi(X) - \max_{j\neq i} e_j^T A\phi(X) - \beta| - (e_i^T A\phi(X) - \max_{j\neq i} e_j^T A\phi(X) - \beta)\} .$$

This function has a unique minimum only for those values of A satisfying (7). Note that

$$J(A) = 0 \quad \text{if} \quad e_i^T A\phi(X) > e_j^T A\phi(X) \quad \text{for all} \quad i \neq j,\ X \in \text{class}\ \omega_i$$
$$= -(e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) - \beta) \quad \text{otherwise} .$$

The gradient of A is

$$\frac{\partial J}{\partial A} = 0 \quad \text{if} \quad e_i^T A\phi(X) > e_j^T A\phi(X) \quad \text{for all} \quad i \neq j,\ X \in \text{class}\ \omega_i$$
$$= -(e_i\phi^T(X) - \max_{i\neq j} e_i\phi^T(X)) \quad \text{otherwise} .$$

Substituting into the general algorithms (GA.3):

(PA.1)
$$A[n] = A[n-1] + \frac{1}{2}\rho\{[e_i\phi^T(X[n-1]) - \max_{j\neq i} e_j\phi^T(X[n-1])] - [e_i\phi^T(X[n-1]) - \max_{j\neq i} e_j\phi^T(X[n-1])]\operatorname{sgn}(\epsilon_{n-1})\}$$
$$\beta[n] = \beta[n-1] + \frac{1}{2}\rho\{[1 - \operatorname{sgn}(\epsilon_{n-1})] + |1 \operatorname{sgn}(\epsilon_{n-1})|\}$$

where $$\epsilon_{n-1} = e_i^T A[n-1]\phi(X[n-1]) - \max_{j\neq i} e_j^T A[n-1]\phi(X[n-1]) - \beta|n-1] .$$

In this algorithm, no correction is made to A if

$$e_i^T A\phi(X) - \max_{j\neq i} e_j^T A\phi(X) > \beta ,$$

which means X is correctly classified. If X is incorrectly classified, A is incremented by

$$\rho(e_i\phi^T(X) - \max_{j\neq i} e_j\phi^T(X)) .$$

That is,

$$A[n] = A[n-1] + \rho(e_j\phi^T(X[n-1]) - \max_{j\neq i} e_j\phi^T(X[n-1])) .$$

The convergence proof of algorithm (PA.1) with $\beta[n] \equiv 0$ for all n is a modified version of Novikoff's convergence proof for perceptrons, the details of which can be found in Blaydon, Appendix II.2 (B3).

2.3.2 The Relaxation Method

Consider the following linear functional.

$$J(A) = \frac{1}{8}((e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) - \beta) - |e_i^T A\phi(X) - \max_{j\neq i} e_j^T A\phi(X) - \beta|)^2 .$$

Thus

$$J(A) = 0 \quad \text{if} \quad e_i^T A\phi(X) > e_j^T A\phi(X) \quad \text{for all} \quad i \neq j,\ X \in \text{class}\ \omega_i$$
$$= \frac{1}{2}(e_i^T A\phi(X) - \max_{j\neq i} e_j^T A\phi(X) - \beta)^2 \text{ otherwise} .$$

The gradient of $J(A)$ is

$$\frac{\partial J}{\partial A} = 0 \quad \text{if} \quad e_i^T A\phi(X) > e_j^T A\phi(X) \quad \text{for all} \quad i \neq j,\ X \in \text{class}\ \omega_i$$
$$= (e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) - \beta)(e_i\phi^T(X) - \max_{i\neq j} e_j\phi^T(X)) \quad \text{otherwise.}$$

Substituting into the general algorithm (GA.3) yields:

$$\text{(PA.2)} \quad A[n] = A[n-1] \quad \text{if} \quad e_i^T A[n-1]\phi(X[n-1]) > e_j^T A[n-1]\phi(X[n-1])$$

$$\text{for all } i \neq j,\ X \in \text{class } \omega_i$$

$$= A[n-1] + \rho\{|\epsilon_{n-1}|[e_i\phi^T(X[n-1]) - \max_{i\neq j} e_j\phi^T(X[n-1])\}$$

$$\text{otherwise}$$

where $\epsilon_{n-1} = e_i^T A[n-1]\phi(X[n-1]) - \max_{i\neq j} e_j^T A[n-1]\phi(X[n-1]) - \beta[n-1]$.

$$\beta[n] = \beta[n-1] \quad \text{if} \quad e_i^T A[n-1]\phi(X[n-1])$$

$$> e_j^T A[n-1]\phi(X[n-1])$$

$$\text{for all } i \neq j,\ X \in \text{class } \omega_i$$

$$= \beta[n-1] + \rho\{|\epsilon_{n-1}| + \epsilon_{n-1}\} \quad \text{otherwise} .$$

Algorithm (PA.2) behaves as algorithm (PA.1) except that the increment added to $A[n]$ is weighted by the magnitude of the absolute error.

2.3.3 The Minimum-Square-Error Method

Consider the following strictly convex and differentiable linear functional.

$$J(A) = \frac{1}{2}(e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) - \beta)^2 .$$

The gradients of $J(A)$ are

$$\frac{\partial J}{\partial A} = (e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) - \beta)(e_i\phi^T(X) - \max_{i\neq j} e_j\phi^T(X))$$

$$\frac{\partial J}{\partial \beta} = -(e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) - \beta) .$$

Substituting into the general algorithm (GA.3) yields:

$$\text{(PA.3)} \quad A[n] = A[n-1] - \rho\{\varepsilon_{n-1}[e_i\phi^T(X[n-1] - \max_{i\neq j} e_j\phi^T(X[n-1])]\}$$

$$\beta[n] = \beta[n-1] + \rho\{|\varepsilon_{n-1}| + \varepsilon_{n-1}\}$$

where $$\varepsilon_{n-1} = e_i^T A[n-1]\phi(X[n-1]) - \max_{i\neq j} e_j^T A[n-1]\phi(X[n-1]) - \beta[n-1] .$$

Algorithm (PA.3) terminates only if

$$e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) = \beta$$

for all N training patterns. Therefore, the existence of a solution for A in (7) does not assure the convergence of the algorithm. After each iteration, the following condition can be checked.

$$e_i^T A\phi(X) - \max_{i\neq j} e_j^T A\phi(X) > 0 \quad \text{for all } i \text{ and } X \in \text{class } \omega_i .$$

If it is satisfied, the problem is solved. If not, the number of misclassified training patterns is counted; the learning procedure is terminated after a fixed number of iterations and the solution with the smallest number of errors is retained.

CHAPTER III

THE MEAN-SQUARE-ERROR CRITERION IN MULTICATEGORY PATTERN RECOGNITION

The unknown parameters in a discriminant function are established by selecting and optimizing a performance criterion. If the criterion is well defined, the parameters can be found by search techniques such as the gradient descent method in (9). For the two-class problem, the criterion function should be chosen so that the discriminant function $f(X)$ has approximately one value whenever X comes from class ω_1 and a different value whenever X comes from class ω_2.

One such method is the mean-square-error criterion, first used by Widrow and Hoff (W2) to find an adaptive procedure for classifying binary patterns. Koford and Groner (K1) later extended Widrow and Hoff's procedure to real-valued patterns and showed that the adaptive pattern classifier converges to the optimal classifier (in the Bayes sense) for normally distributed pattern classes with equal covariance matrices. Sebestyen (S1) proposed a two-step clustering transformation which maps all patterns belonging to one class into the neighborhood of a fixed point. Sebestyen's clustering transformation actually minimizes a mean-square-error criterion.

A mean-square-error approach to the M-class problem, using the generalized inverse idea for rapid computation, was undertaken by Wee (W3). This approach allows numerous results obtained in the

two-class problem to be extended to the M-class problem. In fact, the results show that the pattern classifiers proposed by Widrow and Hoff (W2), Patterson and Womack (P1) and Chaplin and Levadi (C1) are special cases of Wee's procedure.

The remainder of this chapter will contain a mathematical formulation and some applications of the mean-square-error criterion.

3.1 The Mean-Square-Error Criterion in Pattern Recognition

The mean-square-error criterion will be formulated for the two-class problem and then extended to the M-class problem. Since minimizing the mean-square-error does not necessarily minimize the probability of misclassification, the relation between the resulting discriminant function and the optimum Bayes discriminant function will be discussed under special circumstances.

If the discriminant function $f(X)$ in (2) is considered to be a transformation from the feature space to the real line, then a reasonable solution to the two-class pattern recognition problem will be to find an $f(X)$ which transforms all patterns X belonging to class ω_1 as close as possible to some number $\beta_1 > 0$ and which transforms all patterns belonging to class ω_2 as close as possible to $\beta_2 < 0$. Using this reasoning, the mean-square-error criterion can be formulated as follows.

$$J = \frac{1}{N_1+N_2} \sum_{i=1}^{N_1} [f(X_i(1))-\beta_1]^2 + \sum_{i=1}^{N_2} [f(X_i(2))-\beta_2]^2 \qquad (11)$$

where $\{X_1(i), X_2(i), \ldots, X_{N_i}(i)\}$ are the training patterns from class ω_i $(i = 1,2)$. Patterson and Womack (P1, P2) have proposed a modified version of (11),

$$J(C) = \frac{P_1}{N_1} \sum_{i=1}^{N_1} [C^T X_i(1) - C(2/1)]^2 + \frac{P_2}{N_2} \sum_{i=1}^{N_2} [C^T X_i(2) + C(1/2)]^2$$

where P_1, P_2 are the prior probabilities of pattern classes ω_1 and ω_2, respectively; $C(i/j)$ denotes the cost in assigning pattern X to class ω_i when it really belongs to class ω_j. Since $J(C)$ is a convex function of C, a unique minimum exists, which can be obtained by computer search techniques.

Patterson and Womack (P1) also relate the optimum Bayes discriminant function and the least mean-square-error discriminant function as the number of training patterns from each class approaches infinity. This relationship is stated in Theorem 3.1.1.

If $p_i(X)$ denotes the probability density of X, given that class ω_i is active, and P_i is the prior probability of class ω_i, the Bayes discriminant function is:

$$\begin{aligned} \beta(X) &\geq 0 \quad \text{if} \quad X \in \text{class } \omega_1 \\ &< 0 \quad \text{if} \quad X \in \text{class } \omega_2 \end{aligned} \tag{12}$$

where

$$\beta(X) \triangleq \frac{C(2/1)P_1 p_1(X) - C(1/2)P_2 p_2(X)}{P_1 p_1(X) + P_2 p_2(X)} .$$

Theorem 3.1.1. If $\lim_{N_1 \to \infty} \frac{1}{N_1} \sum_{i=1}^{N_1} [C^T X_i(1) - C(2/1)]^2 =$

$E_1[C^T X - C(2/1)]$ almost everywhere

and

$$\lim_{N_2 \to \infty} \frac{1}{N_2} \sum_{i=1}^{N_2} [C^T X_i(2) + C(1/2)]^2 = E_2[C^T X + C(1/2)]^2$$

almost everywhere

then, if $C = C^*$ minimizes $J(C)$ as $N_1, N_2 \to \infty$, C^* also minimizes the following integral

$$\int_{\Omega_X} [C^T X - \beta(X)]^2 [P_1 p_1(X) + P_2 p_2(X)] dX$$

provided the integral exists, where Ω_X is the feature space.

In other words, the least mean-square-error discriminant function is the best mean-square, linear, approximation to the optimum Bayes discriminant function as the number of training patterns from each class approaches infinity.

Equation (11) can be rewritten as

$$J = \frac{1}{N} \sum_{i=1}^{2} \sum_{j=1}^{N_i} [f(X_j(i)) - \beta_i]^2 ,$$

where $N = N_1 + N_2$ or, equivalently, as:

$$J(C) = \frac{1}{N} \sum_{i=1}^{2} \sum_{j=1}^{N_i} (X_j^T(i)C - \beta_i)^2 = \frac{1}{N} \|\Psi C - \beta\|^2 \qquad (13)$$

where

$$\Psi = \begin{bmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_{N_1}^T \\ -X_{N_1+1}^T \\ \vdots \\ -X_N^T \end{bmatrix}$$

$X_1^T, X_2^T, \ldots, X_{N_1}^T$ — training patterns from class ω_1

$-X_{N_1+1}^T, \ldots, -X_N^T$ — training patterns from class ω_2

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix} \quad , \text{ where } \quad \beta_i > 0 \quad \text{for all} \quad i = 1,2,\ldots,N$$

The mean-square-error criterion will now be extended to the M-category problem. Consider the set of M discriminant functions $\{f_i(X)\}_{i=1}^{M}$ as a set of transformations which, for each i, f_i maps all multidimensional patterns belonging to class ω_i as close as possible to some K-dimensional vertex* $e_i = (e_{i1}, e_{i2}, \ldots, e_{i,K})^T$ as defined in Sec. 2.1. The mean-square error criterion can be written as follows.

$$J = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N_i} (X_j^T(i) a_k - e_{ik})^2 \ . \tag{14}$$

This is equivalent to

$$J = \frac{1}{N} \left\| \Psi A^T - E \right\|^2 \triangleq \frac{1}{N} \text{ trace } \{ (\Psi A^T - E)^T (\Psi A^T - E) \} \tag{15}$$

where A is defined in Sec. 2.1 and

$$\Psi = \begin{bmatrix} \Psi[1] \\ \Psi[2] \\ \vdots \\ \Psi[M] \end{bmatrix} \quad \begin{array}{l} \text{training patterns from class } \omega_1 \\ \text{training patterns from class } \omega_2 \\ \vdots \\ \text{training patterns from class } \omega_M \end{array}$$

---

*The number K = M-1 if all the vertices lie on a sphere with centroid at the origin having equal angles formed by the lines joining the vertices and the origin, that is,

$$e_i^T e_j = 1 \quad \text{if} \quad i = j$$
$$= -1/K \quad \text{otherwise}$$

K = M if $e_i^T e_j = 1$ if i = j
$= 0$ otherwise

$$N = \sum_{i=1}^{M} N_i \quad \text{and} \quad \Psi[i] = \begin{bmatrix} X_1^T(i) \\ \vdots \\ X_{N_i}^T(i) \end{bmatrix} \quad \text{for all } i$$

$$E = \begin{bmatrix} e_1^T \\ e_2^T \\ \vdots \\ e_N^T \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1M} \\ e_{21} & e_{22} & \cdots & e_{2M} \\ \vdots & & & \\ e_{N1} & e_{N2} & \cdots & e_{NM} \end{bmatrix}$$

$$\text{where } e_{ij} > 0 \text{ if } X_i \in \text{class } \omega_j$$
$$e_{ij} < 0 \text{ if } X_i \notin \text{class } \omega_j .$$

The pattern recognition problem becomes the problem of selecting A and E to minimize (15). In Sec. 3.2, the relation to Bayes procedure will be discussed.

3.2 The Generalized Inverse Approach in Multiclass Pattern Recognition

Generalized inverse computations can be used to furnish a quick solution under the mean-square-error criterion for M-class problems, (15), for fixed-size training patterns. Since the rows of matrix E in (15) can be interpreted either as reference points (vertices) or as cost vectors, the pattern classifier proposed by Chaplin and Levadi (C1) and the adaptive classifier of Patterson and Womack (P1) are special cases of the generalized inverse approach proposed by Wee (W3). Wee also extended Theorem 3.1.1 to the M-class problem with (15). However, in both of his interpretations, the matrix E is fixed and a solution to (15) is obtained by letting $dJ/dA = 0$.

A modified version of the generalized inverse approach permitting variation in the E matrix, subject to certain constraints, will now be described. In fact, the proposed method resembles the Ho-Kashyap strategy (H2) for the two-class problem. The introduction of the entries of the E matrix as additional parameters in the optimization problem improves the convergence rate of the algorithm.

Since A is unconstrained, the minimum of J in (15) can be obtained by letting $dJ/dA = 0$. This implies that:

$$A^T = (\Psi^T\Psi)^{-1}\Psi^T E = \Psi^{\#} E$$

where $\Psi^{\#}$ is called the generalized inverse (W3).

Let

$$\overline{X[i]} = \frac{1}{N_i}\sum_{j=1}^{N_i} X_j[i]$$

$$\overline{XX^T} = \frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N_i} X_j[i]X_j^T[i]$$

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{M} N_i\overline{X[i]} \quad .$$

Then,

$$A^T = \left(\sum_{i=1}^{M}\sum_{j=1}^{N_i} X_j[i]X_j^T[i]\right)^{-1}\sum_{i=1}^{M}\sum_{j=1}^{N_i} X_j[i]e_i^T$$

$$= N^{-1}(\overline{XX^T})^{-1}\left(\sum_{i=1}^{M} N_i\,\overline{X[i]}\,e_i^T\right).$$

Set

$$e_i^T = (C(1/i),C(2/i),\ldots,C(M/i)) = C^T(i)$$

and assume

$$C(j/i) = 0 \quad \text{if} \quad i = j$$
$$= c > 0 \quad \text{otherwise} \tag{16}$$
$$A^T = c(\overline{X}\, X^T)^{-1}\left(\overline{X} - \frac{N_1}{N}\overline{X[1]},\ \overline{X} - \frac{N_2}{N}\overline{X[2]},\ldots,\overline{X} - \frac{N_M}{N}\overline{X[M]}\right).$$

A reasonable decision rule is:

Decide $X \in$ class ω_i if

$$\|X^T A^T - C^T(i)\|^2 < \|X^T A^T - C^T(j)\|^2 \quad \text{for all} \quad j \neq i. \tag{17}$$

Expanding (17), the decision rule becomes:

Decide $X \in$ class ω_i if

$$c \sum_{\substack{k=1 \\ k\neq i}}^{M} X^T a_k - \frac{1}{2}(M-1)c^2 > c \sum_{\substack{k=1 \\ k\neq j}}^{M} X^T a_k - \frac{1}{2}(M-1)c^2$$

or, if

$$X^T a_j > X^T a_i \quad \text{for all} \quad j \neq i \tag{18}$$

where

$$a_i = c(\overline{XX^T})^{-1}\left(\overline{X} - \frac{N_i}{N}\overline{X[i]}\right).$$

Equation (18) is the result of the generalized inverse approach when the E matrix satisfies (16).

In practice, if a pattern class has multimodal structure, a cluster-seeking technique should be employed and one discriminant function should be assigned to each cluster before applying (18).

The relation between the generalized inverse approach and the optimum Bayes decision rule is as follows. By analogy to (12), the optimum Bayes discriminant functions for the M-class problem are (W4)

$$B_i(X) = \frac{\sum_{k=1}^{M} C(i/k)p_k(X)P_k}{\sum_{k=1}^{M} p_k(X)P_k} \qquad \text{for all } i = 1,2,\ldots,M$$

where, ideally, $B_i(X) < B_j(X)$ if X is from class ω_i.

Let

$$B^T = (B_1(X), B_2(X),\ldots,B_M(X)) .$$

Theorem 3.1.1 can be extended to the M-class problem as follows.

Theorem 3.2.1. If $\lim_{N_i\to\infty} \frac{1}{N_i} \sum_{j=1}^{N_i} \|X_j^T[i]A^T - e_i^T\|$

$$= E_i[\|X^T[i]A^T - e_i^T\|^2] \quad \text{almost everywhere}$$

then

$$\lim_{N_i\to\infty} \sum_{i=1}^{M} \frac{N_i}{N} \left(\frac{1}{N_i} \sum_{j=1}^{N_i} \|X_j^T[i]A^T - e_i^T\|^2 \right)$$

$$= \sum_{i=1}^{M} P_i E_i[\|X^T[i]A^T - e_i^T\|^2] \quad \text{almost everywhere} .$$

Furthermore, minimizing J of (15) is equivalent to minimizing the integral

$$\int_{\Omega_X} \|X^T A^T - B^T\|^2 \left(\sum_{k=1}^{M} p_k(X)P_k \right) dx .$$

The proof is given in Wee (W3).

Thus, the discriminant functions obtained by the generalized inverse approach are closest among all linear functions to optimum Bayes discriminant functions in a mean-square sense as the number of training patterns approaches infinity.

It is assumed that the entries of matrix E can be varied subject to the constraint that any rwo vector e in the grouping

of rows corresponding to class ω_i must satisfy the inequality

$$e^T(n)e_i(0) \geq e^T(n)e_j(0), \quad \text{for all } j \neq i$$

where n is the iteration number, and where $e_i(0)$ is the vertex vector assigned to class ω_i. It satisfies:

$$\begin{aligned} e_i^T(0)e_j(0) &= \alpha \quad \text{if } i = j \qquad \alpha > \beta \\ &= \beta \quad \text{otherwise} \end{aligned}$$

where α, β are real numbers.

The problem becomes to find A and E so as to minimize

$$J(A,E) = \frac{1}{N} \|XA^T - E\|^2 = \frac{1}{N} \operatorname{trace}\{(XA^T-E)^T(XA^T-E)\} .$$

The complete algorithm can be derived by minimizing J with the gradient descent method.

(GA.4)
$$\begin{aligned} A^T[0] &= \psi^{\#}E[0] \\ D[n] &= \psi A^T[n] - E[n] \\ A^T[n+1] &= A^T[n] + \psi^{\#}\delta E[n] \\ E[n+1] &= E[n] + \delta E[n] \end{aligned}$$

where

$$\begin{aligned} \delta E[n]_{ij} &= \rho D[n]_{ij} \quad \text{if } e[n]_{ij}e_j[0] \geq e[n]_{ij}e_\ell[0] \text{ for all } \ell \neq j \\ &= 0 \qquad \text{otherwise} . \end{aligned}$$

Here, $\delta E[n]_{ij}$ denotes the ith row vector in the grouping of rows corresponding to class ω_i. The convergence proof is given in Appendix A. A computer simulation of (GA.4) was performed on CDC 3600. Two artificial pattern recognition problems were considered. The first problem contains three linearly separable

pattern classes with two training patterns from each class. The decision surfaces which correctly classify all training patterns are shown in Figure 2. The second problem contains three linearly unseparable pattern classes. There are eight training patterns from class 1, six from class 2 and two from class 3. The decision surfaces which misclassify one training pattern are shown in Figure 3.

3.3 Some Properties of Least-Mean-Square-Error Pattern Classifiers

The properties of linear two-class pattern classifiers which are based on the mean-square-error criterion have been investigated by a number of authors (K1, P1, P2). In this section, the statistical properties of two-class classifiers will be studied and the results extended to the M-class problem.

The mean-square-error criterion for the two-class problem is given in (13)

$$J(C) = \frac{1}{N} \sum_{i=1}^{2} \sum_{j=1}^{N_i} (X_j^T(i)C - \beta_i)^2$$

or equivalently,

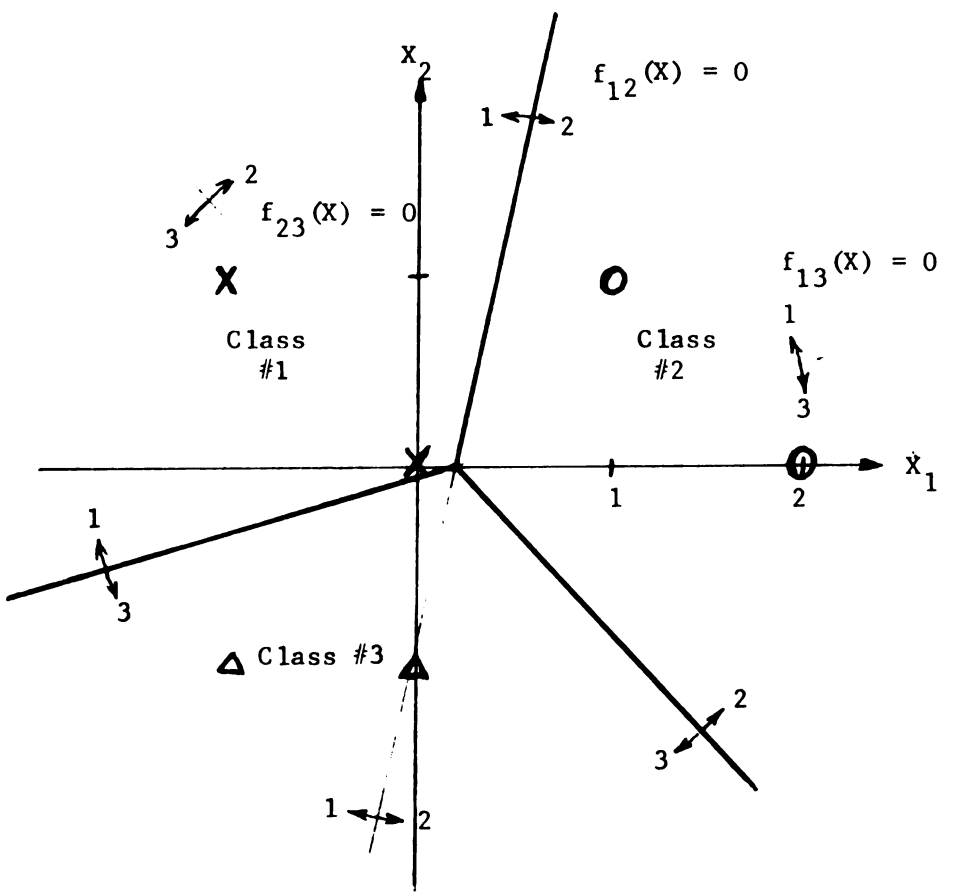
$$J(C) = \frac{1}{N} \sum_{i=1}^{2} \sum_{j=1}^{N_i} (X_j^{*T}(i)C^* + C_0 - \beta_i)^2$$

where $C^* \triangleq (C_1, C_2, \ldots, C_d)$ and $C_0 \triangleq C_{d+1}$.

For simplicity, the star (*) is dropped. Thus

$$J(C) = \frac{1}{N} \sum_{i=1}^{2} \sum_{j=1}^{N_i} (X_j^T(i)C + C_0 - \beta_i)^2 \quad . \tag{19}$$

The problem is to find C and C_0 to minimize $J(C)$.

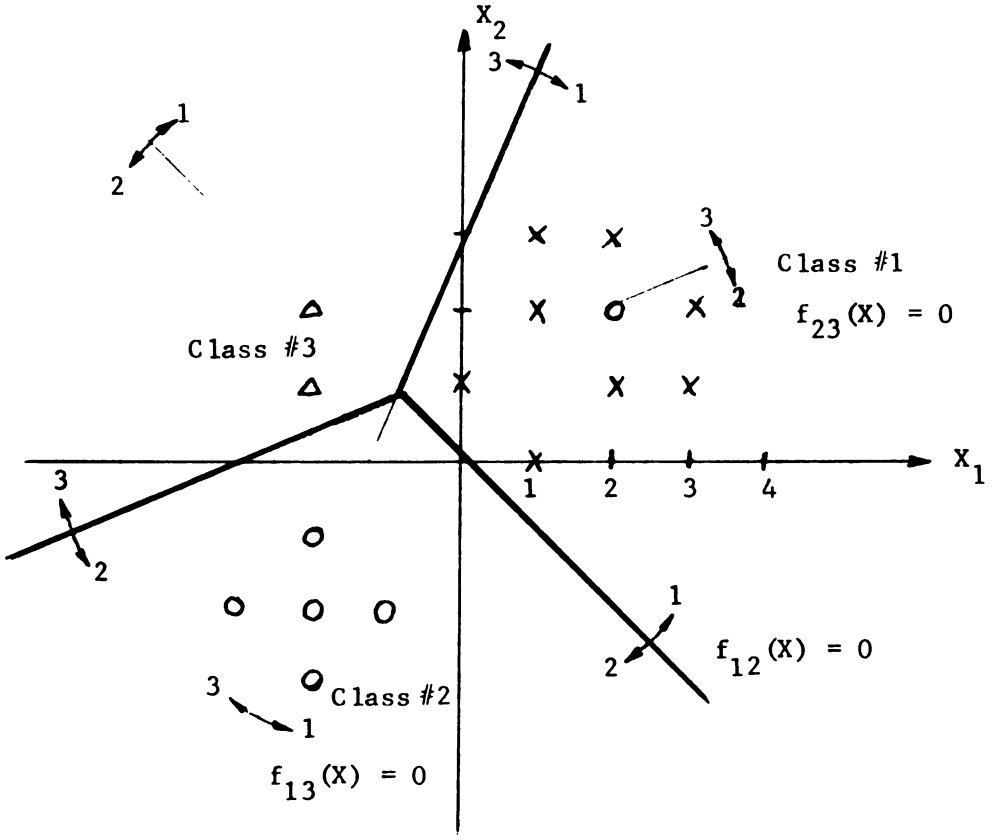


$$f_{12}(X) = X_1 - 0.26X_2 - 0.16$$

$$f_{23}(X) = X_1 + 1.28X_2 - 0.16$$

$$f_{13}(X) = X_1 - 6.4X_2 - 0.16$$

Figure 2. Decision surfaces for algorithm GA.4.



$$f_{12}(X) = X_1 + 1.1X_2 + 0.01$$

$$f_{23}(X) = X_1 - 2.3X_2 + 2.9$$

$$f_{13}(X) = X_1 - 0.3X_2 + 1.2$$

Figure 3. Decision surfaces for algorithm GA.4

Let

$$M_i \triangleq \frac{1}{N_i} \sum_{j=1}^{N_i} X_j(i) \qquad \text{and}$$

$$\Phi_i \triangleq \frac{1}{N_i} \sum_{j=1}^{N_i} (X_j(i) - M_i)\,(X_j(i) - M_i)^T \tag{20}$$

be the sample mean vector and sample covariance matrix for pattern class ω_i.

Equation (19) can be rewritten as

$$J(C) = \frac{1}{N_1+N_2} \sum_{i=1}^{2} N_i \{C^T \Phi_i C + (M_i^T C + C_0 - \beta_i)^2\} . \tag{21}$$

Letting $\partial J/\partial C_0 = 0$,

$$\frac{2}{N_1+N_2} \sum_{i=1}^{2} N_i (M_i^T C + C_0 - \beta_i) = 0$$

or

$$C_0 = \frac{-(M_1 N_1 + M_2 N_2)^T C + N_1\beta_1 + N_2\beta_2}{N_1 + N_2} . \tag{22}$$

If $N_1 = N_2$ and $\beta_1 = 1$, $\beta_2 = -1$, then

$$C_0 = -1/2\ (M_1 + M_2)^T C .$$

Substituting (22) into (21),

$$J(C) = \frac{1}{2} C^T (\Phi_1 + \Phi_2) C + \frac{1}{4} (C^T (M_1 - M_2) - 2)^2 .$$

Let

$$\Phi = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \qquad \text{and} \qquad M_{12} \triangleq M_1 - M_2$$

$$J(C) = C^T \Phi C + \frac{1}{4} (C^T M_{12} - 2)^2 .$$

Letting $\partial J/\partial C = 0$,

$$2\Phi C + \frac{1}{2}(C^T M_{12} - 2)M_{12} = 0$$

or

$$4\Phi C + (M_{12}^T C)M_{12} - 2M_{12} = 0 .$$

Try

$$C = \frac{\Phi^{-1}M_{12}}{2 + \frac{1}{2}M_{12}^T \Phi^{-1}M_{12}} . \tag{23}$$

Thus

$$\frac{4M_{12}}{2 + \frac{1}{2}M_{12}^T \Phi^{-1}M_{12}} + \frac{(M_{12}^T \Phi^{-1}M_{12})M_{12}}{2 + \frac{1}{2}M_{12}^T \Phi^{-1}M_{12}} - 2M_{12}$$

$$= \frac{1}{2 + \frac{1}{2}M_{12}^T \Phi^{-1}M_{12}} [4M_{12} + (M_{12}^T \Phi^{-1}M_{12})M_{12} - 4M_{12}$$

$$- (M_{12}^T \Phi^{-1}M_{12})M_{12}]$$

$$= 0 .$$

From (23),

$$C = K\Phi^{-1}M_{12} = K\Phi^{-1}(M_1 - M_2)$$

where

$$K = \frac{1}{2 + \frac{1}{2}M_{12}^T \Phi^{-1}M_{12}} .$$

The least-square-error linear classifier for the two-class problem is:

$$\begin{aligned} &\text{say } X \in \text{class } \omega_1 \quad \text{if} \quad X^T C + C_0 \geq 0 \\ &\text{say } X \in \text{class } \omega_2 \quad \text{if} \quad X^T C + C_0 < 0 \end{aligned} \tag{24}$$

where C and C_0 are defined in (23) and (22) respectively.

On the other hand, the optimal Bayes pattern classifier for the two-calss problem, assuming normal distribution with equal covariance matrices, is (D2)

$$X \in \text{class } \omega_1 \text{ if}$$

$$X^T \Phi^{-1}(M_1 - M_2) - \frac{1}{2}(M_1 + M_2)^T \Phi^{-1}(M_1 - M_2) \geq 0 \ . \tag{25}$$

Let

$$C = \Phi^{-1}(M_1 - M_2) \quad \text{and} \quad C_0 = \frac{1}{2}(M_1 + M_2)^T C \ .$$

Equation (25) becomes,

$$X \in \text{class } \omega_1 \text{ if}$$

$$X^T C + C_0 \geq 0$$

which is equivalent to (24).

Therefore, the least-mean-square-error classifier is equivalent to the optimal parametric classifier for normally distributed patterns with equal covariance matrices.

A different proof leading to the above result has been provided by Koford and Groner (K1), who also show that a simple modification of the least-mean-square-error adaption procedure enables the adaptive structure to converge to a nearly optimal classifier, even though the numbers of training patterns from the two categories are not equal.

For the M-class problem, the mean-square-error criterion is given in (14).

$$J = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N_i} (X_j^T(i)a_k - e_{ik})^2$$

or equivalently,

$$J = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N_i} (X_j^{*T}(i)a_k^* + a_{ko} - e_{ik})^2$$

where

$$X \triangleq (X^*,1)^T \quad \text{and} \quad a_k \triangleq (a_k^*, a_{ko})^T .$$

Again, for simplicity, the star (*) is dropped. Thus

$$J = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N_i} (X_j^T(i)a_k + a_{ko} - e_{ik})^2 .$$

The problem of minimizing J with respect to a_k and a_{ko} is equivalent to minimizing $J(a_k, a_{ko})$, $k = 1,2,\ldots,M$, where

$$J(a_k, a_{ko}) = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N_i} (X_j^T(i)a_k + a_{ko} - e_{ik})^2 . \tag{26}$$

Applying (20) and assuming $N_1 = N_2 = \ldots = N_M$, Equation (26) can be rewritten as:

$$J(a_k,a_{ko}) = \frac{1}{M} \sum_{i=1}^{M} \left(a_k^T \Phi_i a_k + (a_k^T M_i + a_{ko} - e_{ik})^2 \right) . \tag{27}$$

Letting $\partial J/\partial a_{ko} = 0$,

$$\frac{2}{M} \sum_{i=1}^{M} (a_k^T M_i + a_{ko} - e_{ik}) = 0.$$

Let

$$\bar{M} = \frac{1}{M} \sum_{i=1}^{M} M_i \qquad \text{and} \qquad \bar{e}_k = \frac{1}{M} \sum_{i=1}^{M} e_{ik}$$

then

$$a_{ko} = -a_k^T \bar{M} + \bar{e}_k \ . \tag{28}$$

If $e_i = (e_{i1}, e_{i2}, \ldots, e_{ik})$, $i = 1,2,\ldots,M$, are simplex vertices, then

$$\sum_{i=1}^{M} e_{ik} = 0 \quad \text{for all} \quad k \ .$$

If $e_i = (e_{i1}, e_{i2}, \ldots, e_{iM})$, for all i, are vertices, then $\bar{e}_k \neq 0$.

Substituting (28) into (27),

$$J(a_k, a_{ko}) = \frac{1}{M} \sum_{i=1}^{M} \left\{ a_k^T \Phi_i a_k + \left[(M_i - \bar{M})^T a_k - (\bar{e}_k + e_{ik}) \right]^2 \right\} \ .$$

Setting $\partial J / \partial a_k = 0$,

$$\frac{1}{M} \sum_{i=1}^{M} \Big[2a_k^T \Phi_i + 2a_k^T M_i M_i^T + 2a_k^T \bar{M} \bar{M}^T - 4a_k^T M_i \bar{M} - 2(\bar{e}_k + e_{ik})(M_i - \bar{M})^T \Big] = 0$$

or

$$a_k^T \left[\Phi + \left(\frac{1}{M} \sum_{i=1}^{M} M_i M_i^T - \bar{M} \bar{M}^T \right) \right] = \frac{1}{M} \sum_{i=1}^{M} (\bar{e}_k + e_{ik})(M_i - \bar{M})^T \ .$$

Let

$$T \triangleq \frac{1}{M} \sum_{i=1}^{M} M_i M_i^T - \bar{M} \bar{M}^T$$

and

$$V_k \triangleq \frac{1}{M} \sum_{i=1}^{M} (\bar{e}_k + e_{ik})(M_i - \bar{M})^T .$$

Then

$$a_k^T = V_k[\Phi + T]^{-1} . \tag{29}$$

Substituting (29) into (28),

$$a_{ko} = -V_k[\Phi + T]^{-1}\bar{M} + \bar{e}_k . \tag{30}$$

Equations (29) and (30) indicate the relation between the optimal solution based on the least-mean-square-error criterion and the sample covariance matrices and sample mean vectors.

3.4 Relation Between Linear Inequalities and the Mean-Square-Error Criterion

In the two-class problem, a correct decision is made if

$$yC^TX(i) > 0 \qquad i = 1,2$$

where $y = 1$ if $i = 1$, and $y = -1$ if $i = 2$. Let $Z = yC^TX(i)$. Ideally, C^T is chosen to maximize the probability that $Z > 0$.

On the other hand, the mean-square-error criterion implies that C^T is chosen to map $X(i)$ as close as possible into points ± 1, which minimizes the mean-square-error

$$\overline{|C^TX(i) - y|^2} , \quad \text{or} \quad \overline{(Z-1)^2} .$$

In the M-class problem, the corresponding linear inequality is:

$$y_i AX(i) > 0 \quad \text{for all} \quad i$$

where A is a linear transformation which maps $X(i)$ into the vertices $\{e_1, e_2, \ldots, e_k\}$, $k \leq M$. Let $W = Y_i AX(i)$. Ideally, the matrix A is chosen to maximize the probability that $W > 0$. Chaplin and Levadi (C1) have shown that the result of maximizing the probability that $W > 0$ is identical to that obtained by mini minimizing

$$\overline{|W - e_1|^2} \quad .$$

It can be shown that the result of maximizing the probability that $Z > 0$ is identical to that obtained by minimizing

$$\overline{(Z - 1)^2} \quad .$$

Consider $Z = yC^T X(i)$ as a random variable with unknown density function and finite variance $\sigma_Z^2 = E(Z - \bar{Z})^2$, where $\bar{Z}$ denotes the mean. By the Tchebycheff inequality, (D2)

$$\text{prob}(|Z - \bar{Z}| \geq \bar{Z}) \leq \left(\frac{\sigma_Z}{\bar{Z}}\right)^2 \quad .$$

Since $\text{prob}(|Z - \bar{Z}| \geq \bar{Z}) > \text{prob}(Z < 0)$, minimizing the ratio $\sigma_Z^2/\bar{Z}^2$ minimizes $\text{prob}(Z < 0)$, or maximizes $\text{prob}(Z > 0)$.

$$\frac{\sigma_Z^2}{\bar{Z}^2} = \frac{\overline{Z\,Z^T} - \bar{Z}\,\bar{Z}^T}{\bar{Z}^2} = \frac{\overline{Z\,Z^T}}{\bar{Z}^2} - 1 \quad .$$

In order to minimize $\text{prob}(Z < 0)$, the term $\overline{Z\,Z^T}/\bar{Z}^2$ must be maximized.

$$\frac{\overline{Z\,Z^T}}{\bar{Z}^2} = \frac{C^T \overline{X(i)X^T(i)} C}{(C^T \overline{X(i)y})(\overline{X(i)y}\, C^T)} \triangleq H(C) \quad .$$

Setting $\partial H(C)/\partial C = 0$,

$$\overline{X(i)X^T(i)C} - \frac{C^T\overline{X(i)X^T(i)C}}{C^T\overline{X(i)y}}\ \overline{X(i)y} = 0$$

or

$$C = (X(i)X^T(i))^{-1} K\ \overline{X(i)y} \tag{31}$$

where

$$K = \frac{C^T\overline{X(i)X^T(i)C}}{C^T\overline{X(i)y}}\ ,\quad \text{a real number.}$$

Now, minimize

$$\overline{(Z - 1)^2} = C^T\overline{X(i)X^T(i)C} - 2C^T\overline{X(i)y} + 1\ .$$

Letting $\partial\overline{(Z - 1)^2}/\partial C = 0$,

$$\overline{2X(i)X^T(i)C} - 2\overline{X(i)y} = 0$$

or

$$C = \overline{(X(i)X^T(i))}^{-1}\ \overline{X(i)y}\ . \tag{32}$$

If $K = 1$ in (31), then (31) is identical to (32).

For the M-class problem,

$$\overline{|W - e_1|^2} = \overline{|AX(i) - e_i|^2} = \overline{X^T(i)A^TAX(i) - 2e_i^TAX(i) + 1}$$

$$\frac{\partial\overline{|W - e_1|^2}}{\partial A} = 0,\quad \text{implies}$$

$$\overline{X(i)X^T(i)A^T} - \overline{X(i)e_i^T} = 0$$

or

$$A^T = \overline{(X(i)X^T(i))}^{-1}\ \overline{X(i)e_i^T}\ . \tag{33}$$

By analogy to the two-class problem, consider the random variable $\theta = |W|$ with unknown density function and finite variance σ_θ^2.

$$\frac{\sigma_\theta^2}{\bar{\theta}^2} = \frac{\overline{\theta\,\theta^T}}{\bar{\theta}^2} - 1 = \frac{\overline{|W|^2}}{|\bar{W}|^2} - 1 .$$

Equation (33) can be obtained by minimizing $\overline{|W|^2} / |\bar{W}|^2$.

Letting

$$\frac{\partial}{\partial A} \frac{\overline{|W|^2}}{|\bar{W}|^2} = 0 ,$$

$$\frac{\partial}{\partial A} \overline{|W|^2} - K \frac{\partial}{\partial A} |\bar{W}|^2 = 0 \qquad (34)$$

where

$$K = \overline{|W|^2} / |\bar{W}|^2 .$$

Since

$$\overline{|W|^2} = \overline{|Y_i AX(i)|^2} = \sum_{m=1}^{M} \overline{\left(\sum_{j=1}^{K} \sum_{k=1}^{d+1} y_{mj} a_{jk} x_k(i) \right)^2}$$

$$\frac{\partial \overline{|W|^2}}{\partial A_{qr}} = 2 \sum_{m=1}^{M} \sum_{j=1}^{K} \sum_{k=1}^{d+1} \overline{x_r(i) x_k(i) a_{jk} y_{mj} y_{mq}}$$

which is the (q,r)th component of the matrix

$$\frac{\partial \overline{|W|^2}}{\partial A} = \overline{2X(i)X^T(i)A^T Y_i^T Y_i} = 2\,\overline{X(i)X^T(i)A^T} . \qquad (35)$$

Similarly,

$$|\bar{W}|^2 = \overline{|Y_i AX(i)|}^2 = \sum_{m=1}^{M} \left(\sum_{j=1}^{K} \sum_{k=1}^{d+1} \overline{y_{mj} a_{jk} x_k(i)} \right)^2$$

$$\frac{\partial|\bar{W}|^2}{\partial A_{qr}} = 2 \sum_{m=1}^{M} \overline{x_r(i) \left(\sum_{j=1}^{K} \sum_{k=1}^{d+1} y_{mj} a_{jk} x_k(i) \right) y_{mq}}$$

which is the (q,r)th component of the matrix

$$\frac{\partial|\bar{W}|^2}{\partial A} = 2 \overline{X(i)\bar{W}Y_i} \quad . \tag{36}$$

Substituting (35) and (36) into (34) produces

$$2 \overline{X(i)X^T(i)A^T} - K \overline{2X(i)\bar{W}^T Y_i} = 0$$

or

$$A^T = \left(\overline{X(i)X^T(i)} \right)^{-1} \overline{X(i)K\bar{W}^T Y_i} \; .$$

Choosing $K\bar{W}^T = e_1^T$, then

$$A^T = \left(\overline{X(i)X^T(i)} \right)^{-1} \overline{X(i)e_i^T}$$

which is identical to (33).

CHAPTER IV

MULTICATEGORY PATTERN RECOGNITION USING STOCHASTIC APPROXIMATION TECHNIQUES

In the previous chapters, a class of deterministic pattern classification algorithms was discussed. The pattern recognition problem was solved only with respect to the N training patterns given. No questions related to the generalization problem could be answered. The stochastic, or parametric, approach to pattern recognition (B4, K3, P4, W6, Y1) views the N training patterns as samples from the populations corresponding to the pattern classes. The pattern classes are described by conditional probabilities $P(\omega_i|\mathbf{X})$, the probability that pattern $\mathbf{X}$ belongs to pattern class ω_i. In this chapter the function $f_i(x) = a_i^T\varphi(x)$ will be used as an approximation to $P(\omega_i|X)$. The stochastic problem is then one of selecting a suitable criterion function involving all the available information so that the extremum of the criteria function corresponds to a reasonable approximation to $P(\omega_i|X)$, $i = 1,2,\ldots,M$.

In this chapter, the generalized inverse idea is invoked to form the criterion function. The mathematical formulation of the pattern recognition problem is discussed in Sec. 4.1. In Sec. 4.2, two stochastic algorithms which are similar to the deterministic algorithm of Wee (W3) are proposed. Both schemes use information from all training patterns at each stage of the algorithm. In Sec. 4.3, a stochastic algorithm with a special updating property is proposed. At each iteration, only the information from the

particular pattern presented is utilized. As the number of training patterns increases without bound, the algorithm of Sec. 4.2 is equivalent to the algorithm of Sec. 4.3. In Sec. 4.4, the problem of an occasionally mislabeled training pattern is investigated. The mean square approximation error is defined and its upper bound is computed. A computer simulation of the algorithms is presented in Sec. 4.5.

4.1 Mathematical Formulation

Let there be M pattern classes $\omega_1,\omega_2,\ldots,\omega_M$, any one of which can be active to produce a $d+1$ dimensional (augmented) pattern vector X as shown in Figure 4. The prior probabilities $q_1,q_2,\ldots,q_M$ and the probability densities $p_1(\cdot),p_2(\cdot),\ldots,p_M(\cdot)$ are unknown. However, it is possible to observe a sequence $\{(X_i,y_i),\ i = 1,2,\ldots,N\}$ of training patterns. The correct classification of pattern X_i is denoted by $y_i \in (1,2,\ldots,M)$. The patterns $\{X_i\}$ are chosen independently under probability density $p_k(\cdot)$ and prior probability q_k when $y_i = k$.

The controller has two operating modes described as follows;

1) Training mode: the switch is governed by prior probabilities $q_1,q_2,\ldots,q_M$ and the exact location of the switch is observable

2) Decision mode: Same as the training mode except that the location of the switch is unknown.

We assume that $P(\omega_i|X)$ is approximated by $a_i^T\varphi(X)\ \forall\ i = 1,2,\ldots,M$, where $\varphi^T(X) = (\varphi_1(X),\varphi_2(X),\ldots,\varphi_{d+1}(X))$, and $\{\varphi_i(X)\}_{i=1}^{d+1}$ is a set of known and bounded functions of X; $a_i^T = (a_{i1},a_{i2},\ldots,a_{i,d+1})$, and $\{(a_{ij})|\ i = 1,2,\ldots,M,\ j = 1,2,\ldots,d+1\}$ is a set of constants to be determined.

The training patterns are compiled in a matrix.

$$\Phi_N \triangleq \begin{bmatrix} \varphi^T(X_1) \\ \varphi^T(X_2) \\ \vdots \\ \varphi^T(X_N) \end{bmatrix}$$

where $X_1, X_2, \ldots, X_N$ is a set of N training samples; $N = \sum_{i=1}^{M} N_i$ where N_i denotes the number of training patterns from pattern class ω_i.

The coefficients to be determined are compiled in the A matrix defined in Sec. 2.1. The classifications of the training patterns are summarized in the matrix Z_N.

$$Z_N \triangleq \begin{bmatrix} Z^T(X_1) \\ Z^T(X_2) \\ \vdots \\ Z^T(X_N) \end{bmatrix} = \begin{bmatrix} z_1(X_1) & z_2(X_1) & \cdots & z_M(X_1) \\ z_1(X_2) & z_2(X_2) & \cdots & z_M(X_2) \\ \vdots & \vdots & & \vdots \\ z_1(X_N) & z_2(X_N) & \cdots & z_M(X_N) \end{bmatrix}$$

where $z_j(X_i) = \begin{cases} 1 & \text{if } X_i \in \omega_j \quad (y_i = j) \\ 0 & \text{if } X_i \notin \omega_j \quad (y_i \neq j) \end{cases}$

In general, $E_{Z_j|X}[z_j(X)] = 1 \cdot P[z_j(X) = 1|X] + 0 \cdot P[z_j(X) = 0|X]$

$$= P[y = j|X] = P[\omega_j|X] .$$

The matrix P_N is the following matrix of conditional expectations.

$$P_N \triangleq \begin{bmatrix} P(\omega_1|x_1) & P(\omega_2|x_1) & \cdots & P(\omega_M|x_1) \\ P(\omega_1|x_2) & P(\omega_2|x_2) & \cdots & P(\omega_M|x_2) \\ \vdots & \vdots & & \vdots \\ P(\omega_1|x_N) & P(\omega_2|x_N) & \cdots & P(\omega_M|x_N) \end{bmatrix}$$

The random matrix Z_N can be considered as a noisy "measurement" of P_N since $Z_N = P_N + V_N$ where V_N is the "measurement

noise". This particular formulation is very useful in proving Theorem 4.2.2 later.

4.2 Generalized Inverse Adaptation

The stochastic pattern classification problem can be viewed as the problem of constructing an approximation to a function which can only be measured in the presence of noise. In other words, we are given a random pair (Φ_N, Z_N) which contains all the available information, and requested to determine the unknown parameters in A so that the unknown functions $\{P(\omega_i|X)\}_{i=1}^M$ are approximated by $\{a_i^T\varphi(X)\}_{i=1}^M$ for all i and all patterns X.

The criterion function proposed here involves the available random pair $[\Phi_N, Z_N]$ and is similar to that used with the deterministic generalized inverse approach of Wee (W3).

$$J_N(A) = \frac{1}{N}\|Z_N - \Phi_N A^T\|^2 \triangleq \frac{1}{N}\,\mathrm{Trace}([Z_N - \Phi_N A^T]^T[Z_N - \Phi_N A^T]).$$

The matrix $A^T = A_N^T$ which minimizes $J_N(A)$ can be written as

$$A_N^T = [\Phi_N^T \Phi_N]^{-1}\Phi_N^T Z_N .$$

By the gradient descent procedure, this can be computed iteratively as

$$\text{(SA-1)} \qquad A^T(n+1) = A^T(n) - \rho(n)\Phi_N^T[\Phi_N A^T(n) - Z_N]$$

where n denotes the iteration number, $n = 1,2,3,\ldots$, $A^T(1)$ is an arbitrary $m \times M$ matrix (e.g. $A(1) = \beta I$ where $\beta > 0$ is an small positive number) and $\rho(n)$ is the weighting factors required for stochastic approximation algorithms. The following conditions

are necessary for convergence (B3, G1).

$$\lim_{n\to\infty} \rho(n) = 0, \quad \sum_{n=1}^{\infty} \rho(n) = \infty, \quad \sum_{n=1}^{\infty} \rho^2(n) < \infty .$$

The second order gradient scheme defined by

$$A^T(n+1) = A^T(n) - \rho(n) \left(\frac{\partial^2 J_N(A)}{\partial A^2}\right)^{-1} \left(\frac{\partial J_N(A)}{\partial A}\right)\Big|_{A=A(n)}$$

leads to the stochastic algorithm defined below.

$$\frac{\partial J_N(A)}{\partial A} = \Phi_N^T[\Phi_N A^T - Z_N] \quad \text{and}$$

$$\frac{\partial^2 J_N(A)}{\partial A^2} = \Phi_N^T \Phi_N = \sum_{n=1}^{N} \varphi(X_N)\varphi^T(X_N)$$

Let $R(n) = R(n-1) + \varphi(X_N)\varphi^T(X_n)$, $R(0) = \gamma I$ where γ is an arbitrary small positive number. It has been shown by Ralston (R3) that

$$R^{-1}(n) = R^{-1}(n-1) - R^{-1}(n-1)\varphi(X_n)[\varphi^T(X_n)R^{-1}(n-1)\varphi(X_n) + 1]^{-1}\varphi^T(X_n)R^{-1}(n-1).$$

The stochastic second-order algorithm becomes

$$\text{(SA-2)} \qquad A^T(n+1) = A^T(n) - \rho(n)R^{-1}[N]\Phi_N^T[\Phi_N A^T(n) - Z_N] .$$

In both algorithms, the correction term is proportional to the difference between the ideal probability of classification (i.e., $z_i(X) = 0$ or 1) and the approximated probability of classification (i.e., $a_j^T\varphi(X)$). The factors $\rho(n)\varphi_N^T$ and $\rho(n)R^{-1}[N]\Phi_N^T$ determine how the corrections are to be weighted. At the beginning, the weight is heavy. As this difference is reduced, less and less weight is attached to incoming patterns.

We shall now study the asymptotic properties of generalized inverse adaptation as the number of training patterns becomes infinite.

Theorem 4.2.1 $\lim_{N\to\infty} J_N(A) = J(A)$ with probability one where

$$J(A) \triangleq \sum_{i=1}^{M} q_i E_i[\|Z^T(X) - \varphi^T(X)A^T\|^2] \ . \quad \text{And}$$

$$Z^T(X) \triangleq (z_1(X), z_2(X), \ldots, z_M(X)) \ .$$

Proof:
$$\lim_{N\to\infty} J_N(A) = \lim_{N\to\infty} \frac{1}{N} \|Z_N - \Phi_N A^T\|^2$$

$$= \lim_{N\to\infty} \sum_{k=1}^{M} \sum_{i=1}^{M} \frac{N_i}{N} \left[\frac{1}{N_i} \sum_{\substack{j=1 \\ X_j \in \omega_i}}^{N} (z_k(X_j) - \varphi^T(X_j)a_k)^2\right]$$

$$= \sum_{k=1}^{M} \sum_{i=1}^{M} q_i E[z_k(X) - \varphi^T(X)a_k]^2 \quad \text{(By the strong law of large numbers)}$$

$$= \sum_{i=1}^{M} q_i E[\sum_{k=1}^{M} (z_k(X) - \varphi^T(X)a_k)^2]$$

$$= \sum_{i=1}^{M} q_i E_i[\|Z^T(X) - \varphi^T(X)A^T\|^2] \ .$$

We now show that the criterion function $J(A)$ is a suitable criterion since it is equivalent to a mean-squared criteria function

$$J(A) = \sum_{i=1}^{M} q_i E_i[\|Z^T(X) - \varphi^T(X)A^T\|^2]$$

$$= \sum_{i=1}^{M} q_i E_i E_{Z|i}[\sum_{k=1}^{M} (z_k(X) - \varphi^T(X)a_k)^2]$$

$$= \sum_{i=1}^{M} \sum_{k=1}^{M} q_i E_i[P(\omega_k|X) - 2P(\omega_k|X)\varphi^T(X)a_k + (\varphi^T(X)a_k)^2]$$

$$= \sum_{i=1}^{M} \sum_{k=1}^{M} \{q_i E_i[P(\omega_k|X) - P^2(\omega_k|X)] + q_i E_i[P(\omega_k|X) - \varphi^T(X)a_k]^2\}$$

$$= J_0 + \hat{J}(A)$$

where $J_0 = \sum_{i=1}^{M} \sum_{k=1}^{M} q_i E_i[P(\omega_k|X) - P^2(\omega_k|X)]$ is not a function of the

unknown parameters A.

$$\hat{J}(A) = \sum_{i=1}^{M} \sum_{k=1}^{M} q_i E_i[P(\omega_k|X) - \varphi^T(X)a_k]^2 .$$

Thus, minimizing $J(A)$ corresponds to minimizing $\hat{J}(A)$ which is the mean-square-error approximation criterion.

The solution for the unknown coefficients can be written explicitly as follows.

The matrix $A^T = A_*^T$ which minimizes $J(A)$ is,

$$A_*^T = \{ \sum_{i=1}^{M} q_i E_i[\varphi(X)\varphi^T(X)] \}^{-1} \{ \sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_1|X)], \sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_2|X)], \ldots, \sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_M|X)] \} .$$

The relation $\frac{\partial J(A)}{\partial a_k} = 0$ implies

$$\sum_{i=1}^{M} q_i E_i[0 - 2P(\omega_k|X)\varphi(X) + 2(\varphi^T(X)a_k)\varphi(X)]$$

$$= \sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_k|X) - \varphi(X)\varphi^T(X)a_k] = 0 .$$

Thus, $a_k = \{ \sum_{i=1}^{M} q_i E_i[\varphi(X)\varphi^T(X)] \}^{-1} \{ \sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_k|X)] \}$ $k = 1,2,\ldots,M.$

Theorem 4.2.2 $\lim_{N\to\infty} A_N^T = A_*^T$ with probability one under the following conditions.

1) $\sum_{i=1} q_i E_i[\varphi(X)\varphi^T(X)]$ exists and is positive definite;

2) $\sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_k|X)]$ exist for all $k = 1,2,\ldots,M.$

Proof: $A_N^T = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T Z_N$

Writing $Z_N = P_N + V_N$ as in Sec. 4.1,

$$A_N^T = [\frac{1}{N} \Phi_N^T \Phi_N]^{-1} \frac{1}{N} \Phi_N^T [P_N + V_N]$$

$$\text{(i)}\quad \lim_{N\to\infty}\left[\frac{1}{N}\Phi_N^T\Phi_N\right]^{-1} = \lim_{N\to\infty}\Big\{\sum_{i=1}^{M}\frac{N_i}{N}\Big(\frac{1}{N_i}\sum_{\substack{j=1\\ X_j\in\omega_i}}^{N}\varphi(X_j)\varphi^T(X_j)\Big)\Big\}^{-1}$$

$$= \Big[\sum_{i=1}^{M} q_i E_i[\varphi(X)\varphi^T(X)]\Big]^{-1} \quad \text{By the strong law of large numbers.}$$

$$\text{(ii)}\quad \lim_{N\to\infty}\frac{1}{N}\Phi_N^T P_N = \lim_{N\to\infty}\frac{1}{N}\Big[\sum_{i=1}^{M}\sum_{\substack{k=1\\ X_k\in\omega_i}}^{N}\varphi(X_k)P(\omega_1|X_k),\ldots,\sum_{i=1}^{M}\sum_{\substack{k=1\\ X_k\in\omega_i}}^{N}\varphi(X_k)P(\omega_M|X_k)\Big]$$

$$= \Big[\lim_{N\to\infty}\sum_{i=1}^{M}\frac{N_i}{N}\Big(\frac{1}{N_i}\sum_{\substack{k=1\\ X_k\in\omega_i}}^{N}\varphi(X_k)P(\omega_1|X_k)\Big),\ldots,\lim_{N\to\infty}\sum_{i=1}^{M}\frac{N_i}{N}\Big(\frac{1}{N_i}\sum_{\substack{k=1\\ X_k\in\omega_i}}^{N}\varphi(X_k)P(\omega_M|X_k)\Big)\Big]$$

$$= \Big[\sum_{i=1}^{M} q_i E_i(\varphi(X)P(\omega_1|X)),\ldots,\sum_{i=1}^{M} q_i E_i(\varphi(X)P(\omega_M|X))\Big]$$

$$\text{(iii)}\quad \lim_{N\to\infty}\frac{1}{N}\Phi_N^T V_N = \lim_{N\to\infty}\frac{1}{N}\Phi_N^T[Z_N - P_N]$$

$$= \lim_{N\to\infty}\Big[\frac{1}{N}\sum_{i=1}^{M}\sum_{\substack{j=1\\ X_j\in\omega_i}}^{N}\varphi(X_j)[z_1(X_j^T)-P(\omega_1|X_j)],\ldots,\frac{1}{N}\sum_{i=1}^{M}\sum_{\substack{j=1\\ X_j\in\omega_i}}^{N}\varphi(X_j)[z_M(X_j^T)-P(\omega_M|X_j)]\Big]$$

$$= \Big\{\sum_{i=1}^{M} q_i E_{iz}[\varphi(X)(z_1(X^T)-P(\omega_1|X))],\ldots,\sum_{i=1}^{M} q_i E_{iz}[\varphi(X)(z_M(X^T)-P(\omega_M|X))]\Big\}$$

$$= \Big\{\sum_{i=1}^{M} q_i E_i[\varphi(X)(P(\omega_1|X)-P(\omega_1|X))],\ldots,\sum_{i=1}^{M} q_i E_i[\varphi(X)(P(\omega_M|X)-P(\omega_M|X))]\Big\}$$

$$= (0,0,\ldots,0).$$

Combining these three parts, $\lim_{N\to\infty} A_N^T = A_*^T$ with probability one. Q.E.D.

4.3 A Stochastic Approximation Algorithm with an Updating Property

The method of stochastic approximation proposed by Robbins and Monroe (R2, W5) is designed to find the zero of a regression function $R(x) = EV(x)$, where $V(x)$ is a random function. Sometimes, the R-M method is employed to locate the minimum of $R(x)$ by finding the zero of a regression function $E[V'(x)]$.

To study the multidimensional case, let V be a stationary random vector which is observed at discrete times. The probability distribution of V is unknown. Let H be a matrix of parameters, and let $f(V,H)$ be a real valued function of V and H. Our problem is to find a matrix $H = H_*$ which minimize a regression function

$$R(H) \triangleq E_V\{f(V,H)\} .$$

Since the probability distribution of V is unknown, $R(H)$ cannot be evaluated. However, in practice, if we are given a sequence V_n, $n = 1,2,\ldots$, of independent observations of the random vector V, the matrix H_* can be obtained by iteratively finding the zero of a regression function $\nabla_H R(H) = E_V\{\nabla_H f(V,H)\}$, where ∇_H denotes the gradient operator applied with respect to H.

The following theorem concerning the R-M process for finding A_* has been proved by Gladyshev (G1) and was restated for the multidimensional case by Yau (Y1).

Theorem 4.3.1 Let $\nabla_H f(V,H)$ be a random variable with $\nabla_H R(H) = E_V\{\nabla_H f(V,H)\}$, and let H_* be the (unique) solution of $\nabla_H R(H) = 0$. Choose H_1 arbitrarily and define

(GSA-1) $$H_{n+1} = H_n - \rho_n \nabla_H f(V_n,H_n), \quad n = 1,2,\ldots$$

Then $$P[\lim_{n\to\infty} H_n = H_*] = 1$$

$$\lim_{n\to\infty} E\|H_n - H_*\|^2 = 0$$

provided the following conditions are satisfied.

(1) $\inf_{\epsilon\|H-H_*\|\le\frac{1}{\epsilon}} \langle (H - H_*), \nabla_H R(H)\rangle > 0$ for each $\epsilon > 0$ where

$\|H\| \triangleq (\langle H,H\rangle)^{\frac{1}{2}}$ the norm of a matrix H; $\langle H_1,H_2\rangle \triangleq \text{trace}\{H_1^T H_2\}$ the inner product of matrices H_1 and H_2;

(2) There exists a positive number δ such that for all H

$$E\|\nabla_H f(V,H)\| \le \delta(1+\|H\|^2) ;$$

(3) $\nabla_H f(V_n,H_n)$ is a random variable whose conditional distribution, given $H_1,H_2,\ldots,H_n$, is the same as the distribution of $\nabla_H f(V_n,H_n)$.

(4) $\{\rho_n\}$ is a sequence of positive numbers satisfying the conditions

$$\sum_{n=1}^{\infty} \rho_n = 0 \quad \text{and} \quad \sum_{n=1}^{\infty} \rho_n^2 < \infty.$$

In the following, the criterion function $J(A)$ of Sec. 4.2 will be used to formulate a stochastic approximation algorithm with an updating property.

Lemma 4.3.1 $J(A) \triangleq \sum_{i=1}^{M} q_i E_i[\|Z^T(X) - \varphi^T(X)A^T\|^2] = E[\|Z^T(X) - \varphi^T(X)A^T\|^2]$.

Proof: Since $p(Z,X) = \sum_{i=1}^{M} p(Z,X,\omega_i) = \sum_{i=1}^{M} p(Z,X|\omega_i)P(\omega_i)$

$$= \sum_{i=1}^{M} p(Z,X|\omega_i)q_i$$

then

$$E[\|Z^T(X) - \varphi^T(X)A^T\|^2] = \int \|Z^T(X) - \varphi^T(X)A^T\|^2 p(X,Z)\,dX\,dZ$$

$$= \sum_{i=1}^{M} q_i \int \|Z^T(X) - \varphi^T(X)A^T\|^2 p(X,Z|\omega_i)\,dX\,dZ$$

$$= \sum_{i=1}^{M} q_i E_i[\|Z^T(X) - \varphi^T(X)A^T\|^2] .$$

Since the random variable (X,Z) has the same significance as the random variable V and the matrix A^T is equivalent to H, the stochastic algorithm (GSA-1) can be rewritten as follows.

(GSA-2) $$A_{n+1}^T = A_n^T - \rho_n \nabla_A f(X_n,Z_n,A_n^T), \quad n = 1,2,3,\ldots$$

We now consider the following criterion function.

$$J(A) = E[\|Z^T(X) - \varphi^T(X)A^T\|^2]$$

$$= E[f(X,Z,A^T)]$$

where $f(X,Z,A^T) = \|Z^T(X) - \varphi^T(X)A^T\|^2$.

Taking the gradient,

$$\nabla_A f(X,Z,A^T) = 2\varphi(X)[\varphi^T(X)A^T - Z^T(X)] .$$

By invoking (GSA-2), we obtain the following stochastic approximation algorithm, which has an updating property.

(SP-1) $$A_{n+1}^T = A_n^T + \rho_n \varphi(X_n)[Z^T(X_n) - \varphi^T(X_n)A_n^T] .$$

The sequence A_n^T, $n = 1,2,\ldots$, converges with probability one to the value A_*^T which minimize $J(A)$ and, hence, $\hat{J}(A)$.

To show the convergence of the algorithm SP-1, we need to prove the following lemmas.

Lemma 4.3.2 If $P\{\sum_{i=1}^{M} q_i p_i(X) > 0\} = 1$ and $\{\varphi_i(\cdot)\}_{i=1}^{d+1}$ are linearly independent and continuous functions, then A_*^T exists.

Proof. $J(A) = E[\|Z^T(X) - \varphi^T(X)A^T\|^2]$

$$\frac{\partial J(A)}{\partial A} = 0 \Rightarrow E[2\varphi(X)\varphi^T(X)A_*^T - 2\varphi(X)Z^T(X)] = 0 .$$

Thus $A_*^T = \{E[\varphi(X)\varphi^T(X)]\}^{-1}E[\varphi(X)Z^T(X)]$ and the matrix $E[\varphi(X)\varphi^T(X)]$ has been shown (P3) to be nonsingular; hence A_*^T exists.

The following lemma has been proved (Y1).

Lemma 4.3.3 Let T be a symmetric operator on E^{d+1} with eigenvalues $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_{d+1}$. Then

$$\lambda_1\|A^T\|^2 \le \langle TA^T, A^T\rangle \le \lambda_{d+1}\|A^T\|^2$$

for all $A^T \in L(E^{d+1}, E^M)$ where $L(E^{d+1}, E^M)$ is the set of all linear transformations taking E^{d+1} into E^M and forms a vector space.

Lemma 4.3.4 If $P\{\sum_{i=1}^{M} q_i p_i(X) > 0\} = 1$ and $\{\varphi_i(X)\}_{i=1}^{d+1}$ are linearly independent and continuous functions on the pattern space Ω_X, then there exists a constant k such that

$$k\|A^T - A_*^T\| \leq \langle A^T - A_*^T, \nabla_A J(A)\rangle .$$

Proof: $\nabla_A J(A) = E[\varphi(X)\varphi^T(X)]A^T - E[\varphi(X)Z^T(X)]$

$= \Phi(A^T - \Phi^{-1}B)$ since Φ is nonsingular

$= \Phi(A^T - A_*^T)$ since A_*^T exists

where $\Phi \triangleq E[\varphi(X)\varphi^T(X)]$ a symmetric operator

$B \triangleq E[\varphi(X)Z^T(X)]$.

By invoking Lemma 4.3.3, we have

$$\lambda_1\|A^T - A_*^T\|^2 \leq \langle\Phi(A^T - A_*^T), (A^T - A_*^T)\rangle$$

where λ_1 is the smallest eigenvalue of Φ, $\lambda_1 > 0$. Therefore, $k\|A^T - A_*^T\|^2 \leq \langle\nabla_A J(A), A^T - A_*^T\rangle$. Q.E.D.

Lemma 4.3.5 Let $V = (X,Z)$, $\nabla_A f(V, A^T) = 2\varphi(X)[\varphi^T(X)A^T - Z^T(X)]$. If the matrix $M \triangleq 4E[\|\varphi(X)\varphi^T(X)\|^2]$ exists, there exists a constant $d > 0$ such that for all A^T,

$$E\|\nabla_A f(V, A^T)\|^2 \leq d(1 + \|A^T\|^2).$$

Proof: $\|\nabla_A f(V, A^T)\|^2 = \text{Trace}\{[2\varphi(X)\varphi^T(X)A^T - 2\varphi(X)Z^T(X)]^T[2\varphi(X)\varphi^T(X)A^T - 2\varphi(X)Z^T(X)]\}$

$= 4\text{tr}\{A(\varphi(X)\varphi^T(X))^T(\varphi(X)\varphi^T(X))A^T\}$

$$- 8\mathrm{tr}\{Z_{(X)}\varphi^T(X)\varphi(X)\varphi^T(X)A^T\}$$

$$+ 4\mathrm{tr}\{(\varphi(X)Z^T(X))^T(\varphi(X)Z^T(X))\}$$

$$E\|\nabla_A f(V,A^T)\|^2 = 4\mathrm{tr}\{A\ E[\|\varphi(X)\varphi^T(X)\|^2]A^T\}$$

$$- 8\mathrm{tr}\{E[Z_{(X)}\varphi^T(X)\varphi(X)\varphi^T(X)]A^T\}$$

$$+ 4\mathrm{tr}\{E[\|\varphi(X)Z^T(X)\|^2\}$$

$$= \langle M\ A^T, A^T\rangle - \langle \delta, A^T\rangle + \gamma$$

where $\delta = 8E[Z_{(X)}\varphi^T(X)\varphi(X)\varphi^T(X)]$

$$\gamma = 4\mathrm{tr}\{E[\|\varphi(X)Z^T(X)\|^2\}$$

$$\leq \langle M\ A^T, A^T\rangle + \|\delta\| \cdot \|A^T\| + \gamma$$

$$\leq \lambda_M\|A^T\|^2 + \|\delta\| \cdot \|A^T\| + \gamma$$

where λ_M is the maximum eigenvalue of matrix M and is positive. Choosing $d = \lambda_M + \|\delta\| + \gamma$ we have

$$E\|\nabla_A f(V,A^T)\|^2 \leq d(1 + \|A^T\|^2) \ . \qquad \text{Q.E.D.}$$

Theorem 4.3.2 Let A_*^T be the unique matrix minimizing $J(A)$. Let the sequence of matrices A_n^T, $n = 1,2,\ldots$, be generated by the proposed algorithm (SP-1). If the following conditions are satisfied:

1) $P\{\sum_{i=1}^{M} q_i p_i(X) > 0\} = 1$;

2) $\{\varphi_i(\cdot)\}_{i=1}^{d+1}$ are linearly independent and continuous functions of X ;

3) The matrix $M = 4E\{\|\varphi(X)\varphi^T(X)\|^2\}$ exists;

4) $\{[X_n, Z(X_n)],\ n = 1,2,3,\ldots\}$ is an arbitrary training sequence of independent observations;

5) $\{\rho_n\}$ is a sequence of positive numbers satisfying Condition 4

of Theorem 4.3.1.

Then $P[\lim_{n\to\infty} A_n^T = A_*^T] = 1$

and $\lim_{n\to\infty} E\|A_n^T - A_*^T\|^2 = 0.$

Proof: Invoke Theorem 4.3.1 and utilizing Lemma 2 through 5. Condition 1 of Theorem 4.3.1 is satisfied because of Lemma 4.3.4. Condition 2 of Theorem 4.3.1 is satisfied because of Lemma 4.3.5. Condition 3 of Theorem 4.3.1 is satisfied because of Condition 4 of Theorem 4.3.2. Condition 4 of Theorem 4.3.1 is identical to Condition 5 of Theorem 4.3.2. Q.E.D.

4.4 Sensitivity Study and Error Upper Bound

In a practical situation, some of the patterns used for training might be mislabeled. The causes for mislabeling are plentiful and include typing errors and measurement errors. In this section, we assume that it is possible to observe a sequence of sample patterns $\{(X_i,\hat{y}_i),\ i = 1,2,\ldots,N\}$ where $\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_N$ is a sequence of labeling (or classification) numbers containing some mislabels. Furthermore, we write the probability of mislabeling the origin of pattern X_i as f_i.

Let $y_1,y_2,\ldots,y_N$ be the correctly labeled sequence corresponding to $\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_N$. The probability that $(y_i = k)$ is $P(\omega_k)$ for any i. Then

$$P[\hat{y}_i = y_i,\ y_i = k] = P(\omega_k)(1 - f_i)$$

$$P[\hat{y}_i \neq y_i,\ y_i = k] = P(\omega_k)f_i$$

We now study the sensitivity problem of the proposed algorithm. The following is a generalization of the results obtained in the previous sections.

Define $\hat{Z}^T(X) = (\hat{z}_1(X), \hat{z}_2(X), \ldots, \hat{z}_M(X))$

and $$\hat{z}_j(X) = \begin{cases} 1 & \text{if } \hat{y} = y = j \\ 0 & \text{if } \hat{y} \neq y = j \end{cases}$$

In general

$$E_{z_j|X}[\hat{z}_j(X)] = 1 \cdot P[\hat{z}_j(X) = 1|X] + 0 \cdot P[\hat{z}_j(X) = 0|X]$$

$$= P[\hat{y} = y,\ y = j|X] = P[\omega_j|X](1 - f)$$

$$\hat{Z}_N \triangleq \begin{bmatrix} \hat{Z}^T[X_1^T] \\ \hat{Z}^T[X_2^T] \\ \vdots \\ \hat{Z}^T[X_N^T] \end{bmatrix} = \begin{bmatrix} \hat{z}_1(X_1) & \hat{z}_2(X_1) & \cdots & \hat{z}_M(X_1) \\ \vdots & \vdots & & \vdots \\ \hat{z}_1(X_N) & \hat{z}_2(X_N) & \cdots & \hat{z}_M(X_N) \end{bmatrix}$$

$$\hat{P}_N \triangleq = \begin{bmatrix} P(\omega_1|X_1)(1-f_1) & P(\omega_2|X_1)(1-f_1) & \cdots & P(\omega_M|X_1)(1-f_1) \\ P(\omega_1|X_2)(1-f_2) & P(\omega_2|X_2)(1-f_2) & \cdots & P(\omega_M|X_2)(1-f_2) \\ \vdots & \vdots & & \vdots \\ P(\omega_1|X_N)(1-f_N) & P(\omega_2|X_N)(1-f_N) & \cdots & P(\omega_M|X_N)(1-f_N) \end{bmatrix}$$

As in Sec. 4.1, we can write: $\hat{Z}_N = \hat{P}_N + \hat{V}_N$ where $\hat{V}_N$ is the "measurement noise".

Consider the criterion function

$$\hat{J}_N(A) = \frac{1}{N} \|\hat{Z}_N - \Phi_N A^T\|^2 \triangleq \frac{1}{N} \operatorname{tr}\{[\hat{Z}_N - \Phi_N A^T]^T [\hat{Z}_N - \Phi_N A^T]\}$$

The matrix $A^T = \hat{A}_N^T$ which minimizes $\hat{J}_N(A)$ is

$$\hat{A}_N^T = [\Phi_N^T \Phi_N]^{-1} \Phi_N^T \hat{Z}_N .$$

As in the iterative algorithms (SA-1) and (SA-2), $\hat{A}_N^T$ can be written recursively as

(SA-3) $$A^T(n+1) = A^T(n) - \rho(n)\Phi_N^T[\Phi_N A^T(n) - \hat{Z}_N]$$

(SA-4) $$A^T(n+1) = A^T(n) - \rho(n)R^{-1}[N][\Phi_N A^T(n) - \hat{Z}_N]$$

Theorem 4.4.1. $\lim_{N\to\infty} \hat{J}_N(A) = \sum_{i=1}^{M} q_i E_i[\|\hat{Z}^T(X) - \varphi^T(X)A^T\|^2] \triangleq \hat{J}(A)$.

Proof:
$$\lim_{N\to\infty} \frac{1}{N}\|\hat{Z}_N - \Phi_N A^T\|$$
$$= \lim_{N\to\infty} \sum_{k=1}^{M}\sum_{j=1}^{N} (\hat{z}_k(X_j) - \varphi^T(X_j)a_k)^2$$
$$= \lim_{N\to\infty} \sum_{k=1}^{M}\sum_{j=1}^{M} \frac{N_i}{N}\left[\frac{1}{N_i}\sum_{\substack{j=1 \\ X_j\in\omega_i}}^{M} (\hat{z}_k(X_j) - \varphi^T(X_j)a_k)^2\right]$$
$$= \sum_{i=1}^{M}\sum_{k=1}^{M} q_i E_i(\hat{z}_k(X) - \varphi^T(X)a_k)^2 = \sum_{i=1}^{M} q_i E_i[\|\hat{Z}^T(X) - \varphi^T(X)A^T\|^2]$$

Q.E.D.

We now show that the criterion function $\hat{J}(A)$ is a suitable criterion.

$$\hat{J}(A) = \sum_{i=1}^{M} q_i E_i[\|\hat{Z}^T(X) - \varphi^T(X)A^T\|^2]$$
$$= \sum_{i=1}^{M}\sum_{k=1}^{M} q_i E_i[P(\omega_k|X)(1-f) - 2P(\omega_k|X)(1-f)\varphi^T(X)a_k + (\varphi^T(X)a_k)^2]$$

where f is the probability that X is mislabeled

$$= \sum_{i=1}^{M}\sum_{k=1}^{M} q_i E_i[P(\omega_k|X)(1-f) - P^2(\omega_k|X)(1-f)] + \sum_{i=1}^{M}\sum_{k=1}^{M} q_i E_i[P(\omega_k|X)(1-f) - \varphi^T(X)a_k]^2$$

Thus, minimizing $\hat{J}(A)$ is equivalent to minimizing

$\sum_{i=1}^{M}\sum_{k=1}^{M} q_i E_i[P(\omega_k|X)(1-f) - \varphi^T(X)a_k]^2 \triangleq \hat{\hat{J}}(A)$ in $\frac{\partial \hat{J}(A)}{\partial a_k} = 0$ implies that

$$\sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_k|X)(1-f) - \varphi(X)\varphi^T(X)\hat{a}_k] = 0$$

or

$$\hat{a}_k = \{\sum_{i=1}^{M} q_i E_i[\varphi(X)\varphi^T(X)]\}^{-1} \sum_{i=1}^{M} q_i E_i[\varphi(X)P(\omega_k|X)(1-f)] \quad \forall\, k = 1,2,\ldots,M.$$

Hence we obtain

$$\hat{A}_*^T = (\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_M) = \{ \sum_{i=1}^{M} q_i E_i[\varphi(X)\varphi^T(X)] \}^{-1}$$

$$\cdot [\sum_{i=1}^{M} q_i E_i[\varphi(X) P(\omega_1|X)(1-f), \sum_{i=1}^{M} q_i E_i[\varphi(X) P(\omega_2|X)(1-f), \ldots,$$

$$\sum_{i=1}^{M} q_i E_i[\varphi(X) P(\omega_M|X)(1-f)] .$$

If $f = 0$, $\hat{A}_*^T = A_*^T$ or equivalently $\{\hat{y}\}_{i=1}^M$ is identical to $\{y\}_{i=1}^M$. For $f = 1$, we have $\hat{A}_*^T = 0$ (matrix) which indicates that the unknown parameters A^T cannot be estimated because of lack of available information (i.e. no training samples).

Corresponding to (SP-1), we have the following stochastic approximation algorithm.

$$\text{(SA-5)} \qquad A_{n+1}^T = A_n^T + \rho_n \varphi^T(X_n)[\hat{Z}^T(X_n) - \varphi^T(X_n)A_n^T] .$$

The sequence A_n^T, $n = 1,2,\ldots$, converges with probability one to the matrix $\hat{A}_*^T$.

We shall now define the mean square approximation error and compute the error upper bound. The results are also applicable to the algorithms proposed in Sec. 4.2 and 4.3.

Let $\epsilon_k \triangleq E_X[P(\omega_k|X) - \varphi^T(X)\hat{a}_k]^2$

$$= E_X[P^2(\omega_k|X)] - 2\hat{a}_k^T E_X[\varphi(X) P(\omega_k|X)] + \hat{a}_k^T E_X[\varphi(X)\varphi^T(X)]\hat{a}_k$$

$$= E_X[P^2(\omega_k|X)] - 2\hat{a}_k^T \sum_{i=1}^{M} q_i E_i[\varphi(X) P(\omega_k|X)]$$

$$+ \hat{a}_k^T \sum_{i=1}^{M} q_i E_i[\varphi(X) P(\omega_k|X)(1-f)]$$

$$= E_X[P^2(\omega_k|X)] - \hat{a}_k^T \sum_{i=1}^{M} q_i E_i[\varphi(X) P(\omega_k|X)](1 + f)$$

$$= E_X[P^2(\omega_k|X)] - \hat{a}_k^T E_X[\varphi(X)\varphi^T(X)]\hat{a}_k \left(\frac{1+f}{1-f}\right)$$

$$\leq P_{max} - \hat{a}_k^T E_X[\varphi(X)\varphi^T(X)]\hat{a}_k\left(\frac{1+f}{1-f}\right) \triangleq \epsilon_{kb}$$

where $P_{max} \triangleq \max_{1\leq k\leq M} E_X[P^2(\omega_k|X)]$

and ϵ_{kb} is the upper bound for the mean square error ϵ_k, $k = 1,2,\ldots,M$.

Let ϵ_{kbN} be an estimate of the error upper bound after N training samples. The relation between ϵ_{kbN} and ϵ_{kb} can be stated by the following theorem.

Theorem 4.4.2. $\lim_{N\to\infty} \epsilon_{kbN} = \epsilon_{kb}$ with probability 1 .

Proof: $\epsilon_{kN} = E_X[P(\omega_k|X) - \varphi^T(X)\hat{a}_{kN}]^2$

$$= E_X[P^2(\omega_k|X)] - 2\hat{a}_{kN}^T E_X[\varphi(X)P(\omega_k|X)] + \hat{a}_{kN}^T E_X[\varphi(X)\varphi^T(X)]\hat{a}_{kN}$$

and $\epsilon_{kbN} \triangleq P_{max} - 2\hat{a}_{kN}^T E_X[\varphi(X)P(\omega_k|X)] + \hat{a}_{kN}^T E_X[\varphi(X)\varphi^T(X)]\hat{a}_{kN}$.

Since $\hat{a}_{kN} = \left[\sum_{i=1}^{N} \varphi(X_i)\varphi^T(X_i)\right]^{-1} \begin{matrix} \sum_{i=1}^{N} \varphi_1(X_i)\hat{z}_k(X_i) \\ \sum_{i=1}^{N} \varphi_2(X_i)\hat{z}_k(X_i) \\ \vdots \\ \sum_{i=1}^{N} \varphi_M(X_i)\hat{z}_k(X_i) \end{matrix}$

and $\lim_{N\to\infty} \hat{a}_{kN} = \{E_X[\varphi(X)\varphi^T(X)]\}^{-1} E_X[\varphi(X)P(\omega_k|X)(1-f)$

$$= \hat{a}_k$$

then $\lim_{N\to\infty} \epsilon_{kbN} = \epsilon_{kb}$ with probability one.

Pitt and Womack (P5) have suggested that the value of ϵ_{kbN} can be used to test different sets of $\varphi(X)$ functions to compare relative performance based on the available training samples.

4.5 An Example

A simulation of the proposed algorithm (SP-1) was performed on a IBM 1130 computer. The problem was to classify patterns drawn from three categories with equal a priori probabilities and bivariate Gaussian distributions. Each class has the same covariance matrix $\begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$ but with different means. The mean vectors for pattern classes ω_1, ω_2 and ω_3 are $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\begin{pmatrix} 4 \\ 4 \end{pmatrix}$ and $\begin{pmatrix} 3 \\ -3 \end{pmatrix}$ respectively. Since the optimal Bayesian classifiers for this type of pattern classes are known to be linear, we choose

$$\varphi^T(X) = (x_1, x_2, 1) \ .$$

Figure 5 shows the classification performance of the algorithm (SP-1) after each training group. At the completion of each training, the system was tested by classifying 300 unknown patterns (100 patterns from each classes). The misclassification rate and the corresponding Bayes misclassification rate are shown for comparison. The results indicate that the system error rate approaches to the Bayesian error rate, which is about 0.2, as the number of training samples increases.

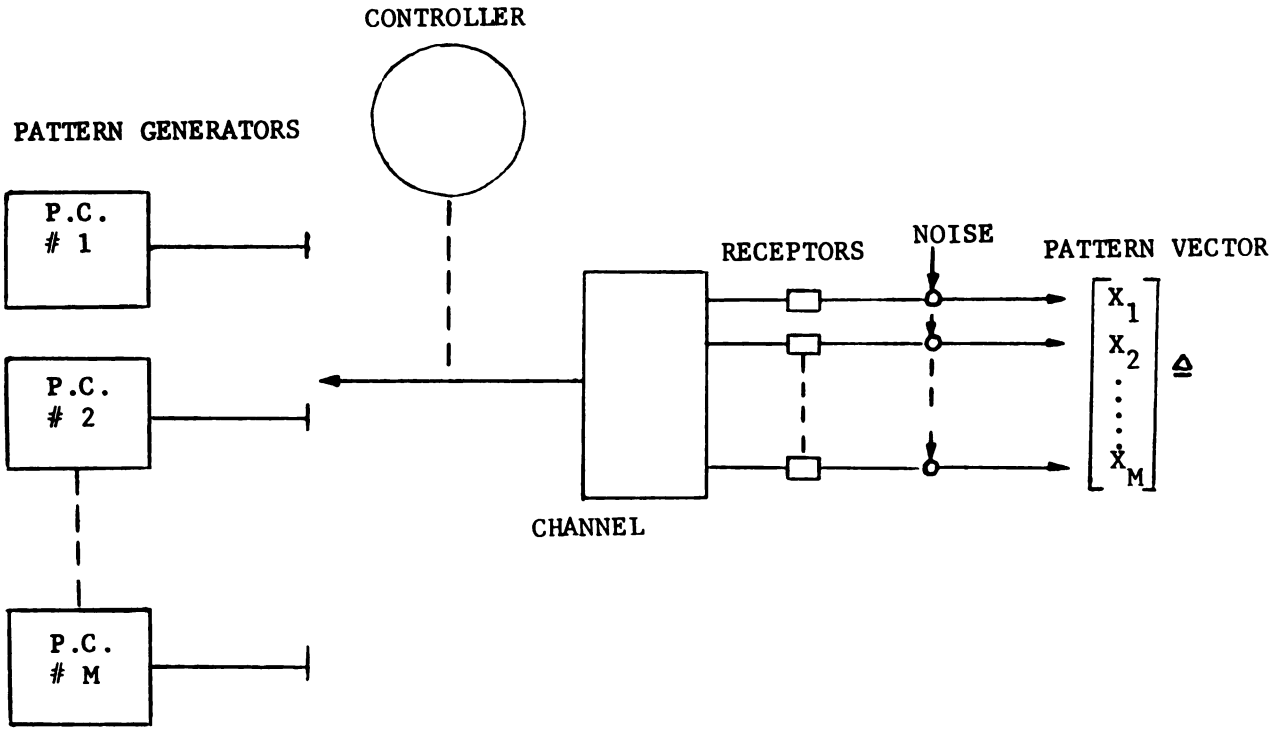


Figure 4. A mathematical model for pattern recognition

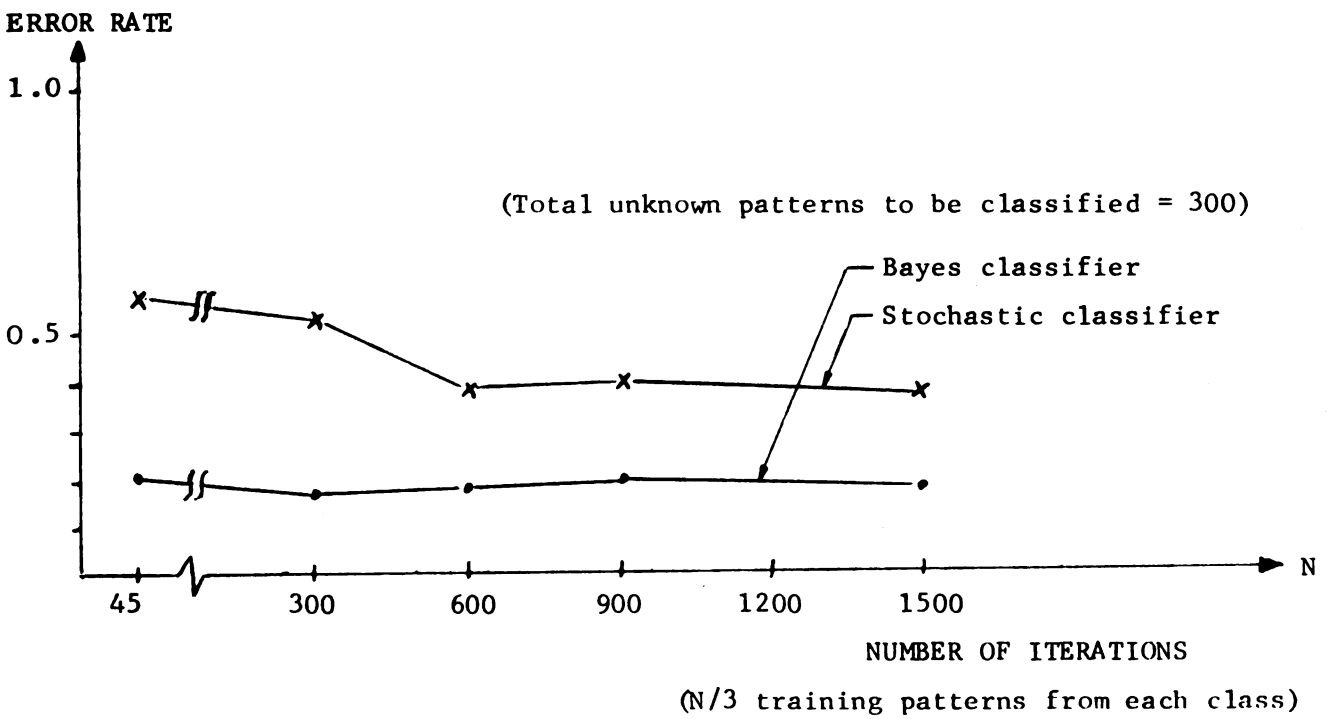


Figure 5. Error rate versus number of iterations.

CHAPTER V

CLUSTER-SEEKING AND PATTERN RECOGNITION

Chapters II, III and IV assumed that the data had structure so the pattern recognition problem could be solved by selecting an algorithm derived according to selected criteria. In reality, information about data structure is rarely known beforehand. Some techniques are thus required to sort the patterns into subsets, such that the patterns in each subset are as much "alike" as possible, and the patterns of different subsets are as much "unalike" as possible. Having obtained the data structure, a pattern classifier can be designed to achieve a minimum number of misclassifications. Some typical cluster-seeking techniques will be discussed in this chapter and a new cluster-seeking procedure will be proposed.

5.1 Some Cluster-Seeking Techniques

Nearly all cluster-seeking techniques were developed after 1960 since they generally require a great deal of computerized high-speed computation. G.H. Ball (B1) classified cluster-seeking techniques into the following six categories: probabilistic, signal detection, clumping, eigenvalue, minimal mode-seeking, and miscellaneous.

Cluster-seeking techniques can also be categorized into the following three groups: (a) Techniques for reproducing "natural" structure. The pattern recognizer selects a grouping criterion;

based on this criterion, the data itself should suggest "natural" clusters. ISODATA by Ball and Hall (B2), and work by Friedman and Rubin (F1), are typical of techniques for reproducing "natural" structure. (b) Techniques for data compression. Patterns are mapped from a higher dimensional space to a lower dimensional space in such a way that the inherent data structure is preserved. These techniques include eigenvalue-type techniques (S1), a non-linear mapping proposed by Sammon (S2), and discriminant analysis (W4). (c) Techniques for minimizing misclassification rate. The probability of error is used as a criterion for clustering. Incorrect classifications imply that new clusters are needed. The algorithm proposed in Sec. 5.2 belongs to this category.

5.1.1. Clustering Techniques for Reproducing "Natural" Structure

Friedman and Rubin (F1) suggested that a real-valued function be defined and evaluated for all possible partitions of the given patterns. The partition which has the maximal function value is selected to represent the data structure.

Suppose a given data matrix Ψ is given in which each row represents a training pattern. A decomposition of the N training patterns into g groups can be achieved by a row partition of Ψ into g submatrices. Without loss of generality, let the sample mean of the N patterns be zero. The total scatter matrix (W4) is given as follows in the notation of Sec. 3.1.

$$T = \Psi^T\Psi = \sum_{i=1}^{N} X_i^T X_i \ .$$

For each partition of the N patterns into g groups, the following matrix identity is satisfied.

$$T = W + B$$

where

$$W = \sum_{k=1}^{g} W_g = \sum_{k=1}^{g} \sum_{\ell=1}^{n_k} [X_\ell(k) - C_k]^T [X_\ell(k) - C_k] .$$

This is called the pooled within-group scatter matrix; n_k is the number of patterns in group k;

$$\sum_{k=1}^{g} n_k = N ;$$

c_k $(k = 1,2,\ldots,g)$ denotes the mean pattern vector of group k; and $X_1(k),\ldots,X_{n_k}(k)$ are the patterns from group k. The matrix

$$B = \sum_{k=1}^{g} n_k C_k^T C_k$$

is called the between-group scatter matrix.

Since the total scatter matrix T is fixed, a natural condition for grouping data is to minimize $|B|$, the determinant of B, or, equivalently, to maximize $|W|$.

One criterion is to maximize

$$\frac{|T|}{|W|} = |I + W^{-1}B| = \prod_{i=1}^{p} (1 + \lambda_i)$$

where λ_i are eigenvalues of $W^{-1}B$. The ratio $|T|/|W|$ is invariant under non-singular linear transformation of the data and is to be maximized over all possible groupings.

The number of possible partitions of N patterns into g groups is enormous; as noted by Friedman and Rubin, the computational problem may be solved in principle but not in practice.

Ball and Hall (B2) proposed a cluster-seeking technique called ISODATA, an adaptive, and semi-heuristic algorithm. The user supplied training patterns and program parameters such as the minimum allowable size of each cluster and the number of clusters desired. Several patterns are initially selected as cluster centers and all patterns are clustered around them by assigning each pattern to the nearest center. New cluster centers are then computed by averaging all patterns in each cluster. Several cycles of assigning patterns to the nearest cluster center and computing new cluster centers are performed. Heuristic subroutines are provided to divide a cluster in two when the number of clusters is small or to combine two of them to reduce the number of clusters.

The main drawbacks of the Ball-Hall technique are that the resulting cluster configuration is highly dependent upon the program parameters supplied by the user, and that there is a lack of good criteria for determining the adequacy of clustering. The type of distance measure also implies spherical or ellipsoidal clusters only.

5.1.2 Clustering Techniques for Data Compression

One clustering algorithm, proposed by Sammon (S2) is basically a nonlinear mapping of the N-dimensional feature space to a low dimensional space (usually two or three dimensions) such that visual identification of clusters is possible. Structure preservation is achieved by moving the N points in the low dimensional space in such a way that the interpoint distances approximate the corresponding interpoint distances in the original high dimensional space.

Let d^*_{ij} be the distance between patterns X_i and X_j where $X_i = (X_{i1}, X_{i2}, \ldots, X_{id})$ denotes a pattern in the original space. Similarly, let d_{ij} be the distance between patterns Y_i and Y_j where $Y_i = (Y_{i1}, Y_{i2}, \ldots, Y_{iD})$ is the pattern in the D-dimensional space corresponding to X_i, $D < d$. An error function for determining the degree of structure preservation is defined as

$$E = \frac{1}{\sum_{i<j}^{N} [d^*_{ij}]} \sum_{i<j}^{N} \frac{[d^*_{ij} - d_{ij}]^2}{d^*_{ij}} .$$

A gradient descent procedure is used to search for a minimum of E by changing the locations of the Y_i's. That is,

$$Y_{id}[n+1] = Y_{id}[n] - \rho \frac{\partial E}{\partial Y_{id}} \Bigg|_{E[n],\, Y_{id}[n]}$$

where n is the iteration number, and $Y_{id}[0]$ is chosen arbitrarily.

The advantages of Sammon's algorithm are (1) that it is free of dependence upon any parameter, and (2) that the resultant configuration in three or less dimensions is easily evaluated by the user. The weaknesses of the algorithm are (1) the large memory requirement and (2) the inaccuracy of the scatter diagram in representing very complex high dimensional data structures.

5.2 A Procedure for Combining Cluster-Seeking and Pattern Classification

A set of weight vectors $\{W_i\}$ defining a set of discriminant functions $\{f_i(X) = W_i^T X\}_{i=1}^{M}$ is needed for non-parametric pattern recognition. A pattern classifier whose discriminant functions

$\{f_i(X)\}$ all have this form is called a linear classifier (N1), for example, the Minimum-Distance Classifier (MDC). A MDC assigns pattern X^* to class ω_i if

$$|X^* - P_i| < |X^* - P_j| \quad \text{for all} \quad \begin{matrix} j = 1,2,\ldots,M \\ i \neq j \end{matrix} \tag{37}$$

where $P_1, P_2, \ldots, P_M$ are cluster points in the feature space which represent the M pattern class; $|X^* - P_i|$ is the Euclidean distance from X^* to the cluster point P_i. Equation (37) can be rewritten as follows:

Let

$$X = (X^*, 1)^T$$

$$W_i = (P_i - 1/2|P_i|^2)^T .$$

Then, say $X^* \in$ class ω_i if

$$W_i \cdot X > W_j \cdot X \quad \text{for all} \quad \begin{matrix} j = 1,2,\ldots,M \\ i \neq j . \end{matrix}$$

Thus, the discriminant functions of a MDC can be written as

$$f_i(X) = W_i \cdot X \quad \text{for all} \quad i .$$

One of the shortcomings of a linear classifier is that the decision regions are necessarily convex. If the pattern classes are multimodel, convex decision regions are certainly inadequate. Non-convex decision regions can be achieved by defining discriminant functions having the form:

$$f_i(X) = \max_{1 \leq m < L_i} \{W_{im} \cdot X\} \quad \text{for all} \quad i . \tag{38}$$

This can also be stated in terms of cluster points P_{ij} as:

assign X^* to ω_i if

$$\min_j |X - P_{ij}| < \min_j |X - P_{kj}| \quad \text{for all} \quad k \neq i$$

where W_{im} is the weight vector for subclass m of class ω_i, and P_{ij} is the cluster point for subclass j of class ω_i; L_i is the number of subclasses for pattern class ω_i. A classifier which implements Equation (38) is called a Piecewise Linear Classifier (PWLC). The subsidiary discriminant function $f_{im}(X)$ is defined as

$$f_{im}(X) = W_{im} \cdot X \quad \text{for all} \quad i = 1,2,\ldots,M \quad m = 1,2,\ldots,L_i \ .$$

If the information about subclasses is given beforehand, the problem can be treated as a multiclass pattern recognition problem. Then, linear classifiers assign pattern X to ω_i if

$$f_{im}(X) > f_{jn}(X) \quad \text{for all} \quad (j,n) \neq (i,m) \ .$$

Therefore, a PWLC is actually a multiclass linear classifier and can be trained by the same methods used to train linear classifiers.

In order to obtain a set of weight vectors for a PWLC, information is needed about the structure of the subclasses. Since such information is rarely known beforehand, structure analysis must be applied; that is, a similarity measure must be selected and the patterns from each pattern class must be partitioned into subclasses. An algorithm will now be proposed for determining a PWLC without prior knowledge of pattern class distributions. The

algorithm combines M-class, linear-classifier training procedures and clustering-seeking techniques under the control of a minimum error probability performance criterion.

The flow chart of the proposed algorithm is given in Figure 6; the main steps of the algorithm are described as follows.

(A) Phase I

Step 1. The iteration number is n. Determine a set of $S_n (S_0 = M)$ discriminant functions, one for each of the S_n nonempty subsets, or clusters, of patterns.

Step 2. Classify each training pattern.

Step 3. Compute the misclassification percentage, $P_r(n)$. If $P_r(n) \leq N_\epsilon$ is the maximum acceptable misclassification percentage for a particular problem, or the total number of discriminant functions exceeds the allowable number of clusters then stop Phase I of the algorithm and begin Phase II. If $P_r(n) > N_\epsilon$ and the total number of discriminant functions is small enough then continue to step 4.

Step 4. Divide each of the S_n clusters into two clusters, one containing all correctly classified patterns and the other containing all misclassified patterns. Learn a new set of discriminant function, one for each cluster. Check to see whether the current set of

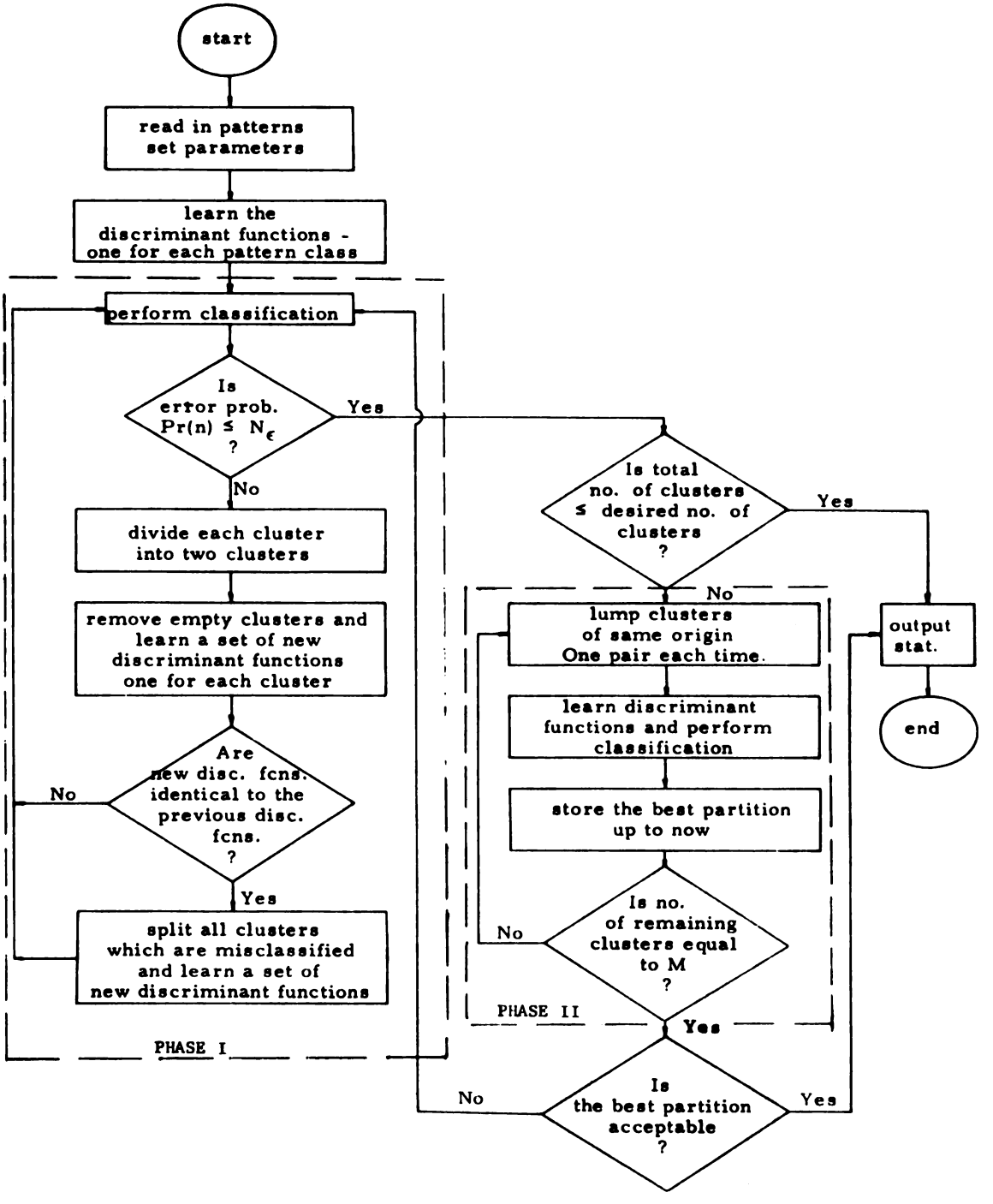


Figure 6. Flow chart of the proposed clustering algorithm.

discriminant functions is identical to the previous set of discriminant functions. If so, split those clusters which were misclassified; let $n = n+1$ and return to step 1.

(B) Phase II

Step 5. Find that pair of clusters from the clusters belonging to each pattern class which are closest together, using an appropriate distance measure. Lump them and compute a new set of discriminant functions. Compute $P_r(n)$ and repeat the procedure until the number of clusters equals the number of categories plus one.

Step 6. Select the best partition. If the partition is acceptable, stop; otherwise, continue to step 2 of Phase I.

Step 7. Terminate the algorithm when the current best partition is inferior to the previous best partition (step 6).

The convergence of the algorithm is intuitively clear, since in Phase I there is a finite number of training patterns; error-free classification can always be achieved if each pattern is considered to be a cluster. In reality, it can be reasonably assumed that the training patterns from each class are somewhat clustered together. Therefore, the number of clusters should be much smaller than the total number of training patterns. Only a finite number of clusters is generated in Phase I. The lumping procedure will

terminate when the number of clusters decreases to $M + 1$. The interaction between Phase I and Phase II provides a search for optimal results.

If a minimum distance classifier (MDC) is employed for classifying training patterns and $N = 0$, the proposed algorithm is actually another version of the condensed nearest neighbor rule (H4). The resulting set of cluster points correctly classify all training patterns. Cover (C2) has shown that the error probability of classifying new patterns is bounded above by twice the Bayes probability of error. If other M-class algorithms are selected for classifying patterns, the proposed algorithm is basically a training procedure for a PWLC so that a bank of subsidary functions for each class can be achieved at the completion of the algorithm.

5.3 A Procedure for Unsupervised Structure Analysis

In supervised learning, a set of training patterns is provided in which each pattern has a label indicating the pattern class to which it belongs. Since no such labels exist in the unsupervised learning problem, the training patterns must be studied for natural groupings. Each such grouping, or cluster, is viewed as defining a pattern class and is assigned a discriminant function. A partition of the N training patterns into M groups, or clusters, is desired for which the misclassification probability and the number of clusters, or discriminant functions, are both minimized. Such a partition converts unclassified patterns into classified training patterns since all patterns in each cluster are tentatively

given the same pattern class label. The algorithms previously defined will then produce discriminant functions for any partition.

The computational problem of searching all possible partitions of N patterns into M groups to find the "best" partition has been solved in principle but not in practice. A suboptimal procedure for searching through the partitions follows.

Any prior knowledge about data structure is inserted into the initial partition. If prior knowledge is not available, the center of gravity for all training patterns can be computed; this center and the (M-1) patterns furthest away from it can be selected as initial cluster centers. The initial partition is then obtained by assigning each pattern to the closest cluster center.

Rubin (R1) proposed an algorithm for finding a local extremum of a criterion function that searched only some of the possible groupings. Starting with the best partition achieved, a single pattern is moved into every group other than its own. If no move increases the criterion, the group label of the pattern is not changed. Otherwise, the pattern is transferred to the group which maximizes the criterion function. This operation is repeated with each pattern in the group. It is then repeated for all patterns in the group with the closest cluster center. After several passes, a point is reached at which changing the label on any single pattern degrades the criterion. This grouping provides a local maximum.

Once the patterns have been tentatively converted into training patterns by labelling each with a group identifier, the algorithm of Sec. 5.2 will produce the error rate (based on the

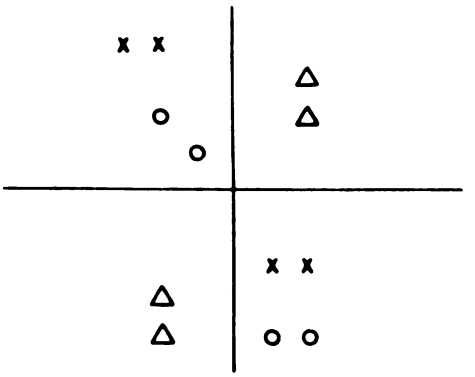
assumed partition) as well as discriminant functions.

5.4 Computer Simulations

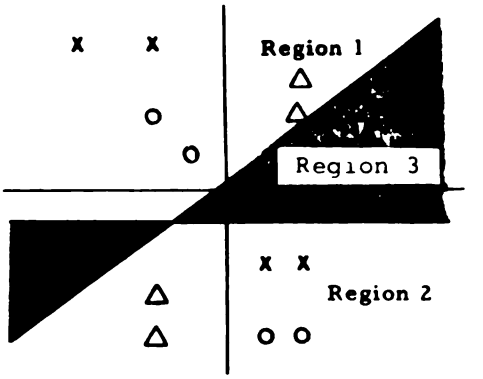
This section contains results obtained by using the CDC 3600 computer system at Michigan State University, to process both the artificial data and the practical data. A FORTRAN listing is provided in (H5).

The artificial pattern recognition problem is shown in Figure 7. The minimum distance classifier (MDC) is selected for classifying training patterns. The data consist of twelve patterns, four for each of 3 pattern classes (Fig. 7a). The patterns are two-dimensional to produce meaningful plots. The initial number of misclassifications is eight (Fig. 7b). That is, assume that each pattern class forms one cluster. The number of misclassified patterns at the fourth iteration before splitting is two (Fig. 7c). At the end of Phase I, there are eight clusters and the number of misclassifications is zero (Fig. 7d). After the completion of Phase II, the best partition of the data is into six clusters and the corresponding number of misclassified patterns is zero (Fig. 7e).

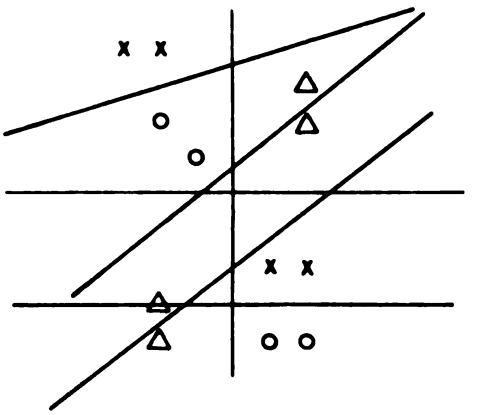
The practical data is the Iris data which was published by Fisher (F2) and repeated by Kendall (K4). There are three species of Iris, setosa, versicolor and virginica. There are fifty flowers of each species and four measurements (sepal length, sepal width, petal length and petal width) are taken on each flower. We labeled them from 1-50, 51-100, and 101-150 respectively. Fisher (F2) constructed a linear function of the four variables to classify them. He found that setosas could be separated from the other two species



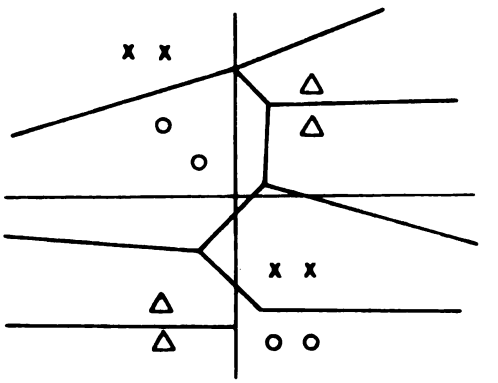
(a) Feature space showing four patterns from each of three classes.



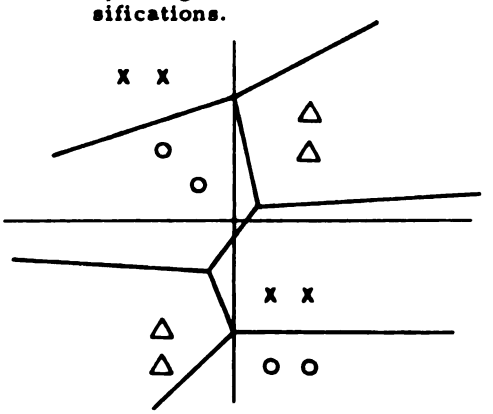
(b) Initial decision regions. Eight patterns misclassified.



(c) Decision regions before splitting. Two misclassifications.



(d) Decision regions before lumping. No misclassifications.



(e) Best partition. No misclassifications.

Figure 7. Example of Algorithm in Figure 6.

perfectly. However, versicolors and virginicas overlapped. Kendall (K4) applied a clustering technique based on a pairwise distance function utilizing only the rank order of the measurements and he was able to separate setosas from others. However, for versicolors and virginicas, he classified 87 flowers and left 13 flowers doubtful. Friedman and Rubin (F1) applied the min trace W criterion to this data. They found when partitioned into three groups, setosas, appeared as a separate group with the other two groups being predominantly versicolors and virginicas except for about ten flowers which were misclassified.

A computer simulation of the system proposed here using minimum distance classifiers has been completed. The initial number of misclassifications was eleven (assuming that each pattern class formed one cluster). With 30 allowable number of clusters and $N_\varepsilon = 0$, the number of misclassified flowers at the end of Phase I was twenty-one. There were thirty-three clusters. Setosas appears as one cluster. Versicolors contains seventeen clusters with fourteen flowers misclassified. Virginicas contains fifteen clusters with seven flowers misclassified. After the completion of Phase II (lumping operation), the best partition of the data was into six clusters and the corresponding number of misclassified flowers was six. The six clusters were: one for setosas, two for versicolors and three for virginicas.

CHAPTER VI

CONCLUSIONS

This chapter discusses the general results of this thesis and possible extensions of this work.

6.1 Thesis Results

There are two sets of results in this thesis.

(1) The development of a class of nonparametric, multiclass pattern recognition algorithms.

The idea of translating a problem into an optimization problem in which search techniques are employed to find the minimum of a linear functional is used in many branches of applied mathematics. Applied to the abstraction problem, this idea provides a great deal of freedom when creating new algorithms. In Chapter II and III, although we make no assumptions about the data structure and pattern classes, the concept of linear inequalities and the mean-square-error criterion were employed to formulate linear functionals. The general multiclass learning algorithms (GA.2) and (GA.3) derived in Chapter II led to particular algorithms such as the fixed increment method (PA.1), the relaxation method (PA.2) and the minimum square error method (PA.3). These algorithms are generalized versions of algorithms in the literature. Chapter III showed that the mean-square-error criterion is equivalent to the

generalized inverse approach. A multiclass algorithm, (GA.4), utilizing the generalized inverse approach is proposed and is unique with this thesis. The convergence proof is given in Appendix A and the utility of the algorithm was demonstrated by a digital computer simulation.

Since no assumptions were made about data structure in Chapters II and III, the pattern recognition problem was solved only for the N given training patterns. In order to investigate the generalization problem, we must view the N fixed training patterns as a sample from a population as was done in Chapter IV. We assumed the existence of probability densities of unknown functional form so the optimal discriminant functions involved conditional probabilities. Based on the available information, the method of stochastic approximation was employed to estimate the unknown discriminant functions and two stochastic algorithms were derived from the generalized inverse approach. Both schemes used information from all training patterns to update the discriminant functions at each iteration. The asymptotic properties of both algorithms are studied for the first time in Theorem 4.2.1 and 4.2.2. In Sec. 4.3, a new stochastic algorithm with an updating property was proposed. Since only the information from the particular pattern presented was utilized at each iteration, the problem of storing large numbers of patterns was eliminated. The convergence of the proposed algorithm was proved by involving Gladyshev's Theorem (G1). In fact, the algorithms of Sec. 4.2 are equivalent to the algorithm in Sec. 4.3 as the number of training samples approaches infinity.

In practical situations, some of the training patterns might be mislabeled. This sensitivity problem is studied for the first time in Sec. 4.4. The results show that the estimated coefficients of the discriminant functions under mislabeled condition are equal to the probability of correct labeling times the estimated coefficients under ideal situation. The mean square approximation error was defined and its upper bound was derived. An estimate of the error upper bound after N training samples can be used as a guide for selecting the $\varphi(X)$ functions.

(2) The introduction of a procedure that combines cluster-seeking and multiclass pattern recognition into a workable pattern recognition system.

In some cases, the pattern class has a multimodal structure; both the number and location of the modes are unknown. In order to minimize the misclassification error, a cluster-seeking technique should be applied to learn the data structure and one discriminant function should be assigned to each cluster (or subclass). A new algorithm for solving the cluster-seeking and multiclass pattern classification problems in a step-wise fashion was proposed in Chapter V. The procedure proposed in Sec. 5.2 exploited structural information to construct discriminant functions. The success of the discriminant functions in classifying training patterns then provided clues about data structure. The procedure was implemented on the CDC Computers at Michigan State University and tested with the problem reported in Sec. 5.4.

The focal point of the thesis is the presentation of a computationally feasible solution to a very difficult, but very

common, pattern recognition problem. The difficulty is caused by the lack of prior knowledge about the data structure, or one's unwillingness to make assumptions about pattern class distributions or linear separability among pattern classes. This problem is extremely acute in situations involving large numbers of pattern and features. The abstraction and clustering problems are attacked simultaneously in Chapter 5 of this thesis. The algorithms proposed derive a piece-wise linear pattern classifier in an iterative manner. Information about structure is incorporated into the classifier. In turn, the operation of the classifier provides structural information.

6.2 Possible Extensions

To simulate the proposed pattern recognition system on a digital computer, the minimum distance classifier was selected for classifying training patterns. In fact, all multiclass algorithms listed in this thesis could be used for classifying patterns. This fact generates a class of recognition procedures and each procedure needs a full-scale test and optimization of program parameters.

For a given pattern recognition problem, a complete exploration of all possible procedures and selection of the "best" is possible in principle but not in practice. However, if one is willing to make assumptions about the pattern classes, he might derive some theoretical results or heuristic rules which optimize the searching among possible procedures.

Another possible extension is to develop an algorithm for unsupervised structure analysis. Since the true classifications of

the patterns are unknown in an unsupervised learning problem, the unclassified patterns must somehow be converted into classified training patterns. Again, if one is willing to impose some probabilistic structure (such as the mixture probability densities), he has converted the problem of clustering into the problem of unsupervised estimation (P6, P7).

REFERENCES

REFERENCES

B 1 G.H. Ball, "Data analysis in the social science; what about the details?", Fall Joint Computer Conference (1965).

B 2 G.H. Ball and D.J. Hall, "ISODATA, a novel method of data analysis and pattern recognition," Technical Report, Stanford Research Institute, Menlo Park, California (1965).

B 3 C.C. Blaydon, "Recursive algorithms for pattern classification," Technical Report No. 520, Division of Engineering and Applied Physics, Harvard University, Cambridge, Massachusetts, March, 1967.

B 4 Colin Blaydon and Yu-chi Ho, "On the abstraction problem in pattern classification," Proc. 1966 NEC, Vol. 22, 857-862.

C 1 W.G. Chaplin and V.S. Levadi, "A generalization of the linear threshold decision algorithm to multiple classes," Proc. Second Symposium on Computer and Information Sciences at Batelle Memorial Institute, August 22-24 (1966).

C 2 T.M. Cover and P.E. Hart, "Nearest neighbor pattern classification," IEEE Trans. on Information Theory, Vol. IT-13, No. 1, January 1967.

D 1 I.P. Devyaterikov, A.I. Propoi, and Y.Z. Tsypkin, "Iterative learning algorithms for pattern recognition," Automation and Remote Control (English Translation), 108-117 (1967).

D 2 R.C. Dubes, The Theory of Applied Probability, (Prentice-Hall, Inc., New York, 1968).

F 1 H.P. Friedman and J. Rubin, "On some invariant criteria for grouping data," J. Amer. Statist. Assoc. 62, 320, 1159-1178 (1967).

F 2 R.A. Fisher, "Multiple measurements in taxonomic problems," Annals of Eugenics, Vol. VII, part II, 179-188, 1936.

G 1 E.E. Gladyshev, "On stochastic approximation," Theory of Probability and Its Applications, Vol. 10, 275-278, 1965.

H 1 Y.C. Ho and A.K. Agrawala, "On pattern classification algorithms: Introduction and survey," Proc. IEEE 56, 12, 2101-2114 (1968).

H 2 Y.C. Ho and R.L. Kashyap, "An algorithm for linear inequalities and its applications," IEEE Trans. Electr. Comp. EC-14, 5.

H 3 Y.C. Ho and R.L. Kashyap, "A class of iterative procedures for linear inequalities," J. SIAM Control 4, 1 (1966).

H 4 P.E. Hart, "The condensed nearest neighbor rule," IEEE Trans. on Information Theory, Vol. IT-4, No. 3, May, 1968.

H 5 A.Y. Hung and R.C. Dubes, "An introduction to multiclass pattern recognition in unstructured situations," Interim Report No. 12, Contract No. AFOSR-1023-67B, Div. Engineering Research, Michigan State University, 1970.

K 1 J.S. Koford and G.F. Groner, "The use of adaptive threshold element to design a linear optimal pattern classifier," IEEE Trans. Information Theory, 12, 1 (1966).

K 2 R.E. Kalman and J.E. Bertram, "Control system analysis and design via the 'second method' of Lyapunov, II Discrete-time system," Trans. ASME J. Basic Engr. 82, D, 2, 371-400 (1960).

K 3 R.L. Kashyap and C.C. Blaydon, "Recovery of function from noisy measurements taken at randomly selected points and its application to pattern classification," IEEE Proc., Vol. 54, 1127-1129, August, 1966.

K 4 M.G. Kendall, "Discrimination and Classification," pp. 165-185, Multivariate Analysis, edited by P.R. Krishnaiah, Academic Press, N.Y., 1966.

N 1 N.J. Nilsson, "Learning Machines," (McGraw-Hill, New York, 1965).

N 2 G. Nagy, "State of the art in pattern recognition," Proc. IEEE, 56, 5 (1968).

P 1 J.D. Patterson and B.F. Womack, "An adaptive pattern classification system," IEEE Trans. Systems Science and Cybernetics SSC-2, 1 (1966).

P 2 J.D. Patterson, T.J. Wagner and B.F. Womack, "A mean-square performance criterion for adaptive pattern classification systems," IEEE Trans. Automat. Control AC-12, 2, 195-197 (1967).

P 3 J.D. Patterson, T.J. Wagner and B.F. Womack, "A performance criterion for adaptive pattern classification systems," Proc. 1966 Joint Automatic Control Conf. (Seattle, Wash.), 38-46.

P 4 J.M. Pitt and B.F. Womack, "A sequentialization of Patterson classifier," Proc. IEEE (letter), Vol. 54, 1987-1988, December, 1966.

P 5 James M. Pitt and Baxter F. Womack, "Additional features of an adaptive, multicategory pattern classification system," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-5, No. 5, July, 1969.

P 6 E.A. Patrick and J.P. Costello, "On unsupervised estimation algorithms," IEEE Trans. on Information Theory, Vol. IT-16, No. 5, September, 1970.

P 7 E.A. Patrick, "On a class of unsupervised estimation problems," IEEE Trans. on Information Theory, Vol. IT-14, No. 3, May, 1968.

R 1 J. Rubin, "Optimal classification into groups: An approach for solving the taxonomy problem," J. Theor. Biol. 15, 103-144 (1967).

R 2 H. Robbins and S. Monroe, "A stochastic approximation method," Ann. Math. Statist., Vol. 23, 462-466, 1952.

R 3 A. Ralston, "A First Course in Numerical Analysis," McGraw-Hill, N.Y., 1965.

S 1 G.S. Sebestyen, "Decision-Making Processes in Pattern Recognition," (The Macmillan Company, New York, 1962).

S 2 J.W. Sammon, Jr., "A nonlinear mapping for data structure analysis," IEEE Trans. Computers, C-18, 401-409 (1969).

W 1 D.V. Widder, "Advanced Calculus," 2nd edition (Prentice-Hall, Inc., New York, 1961).

W 2 B. Widrow and M.E. Hoff, trans., "Adaptive switching circuits," 1960 IRE WESCON Conv. Record, part 4, 96-104 (1960).

W 3 W.G. Wee, "Generalized inverse approach to adaptive multi-class pattern classification," IEEE Trans. Computers C-17, 12 (1968).

W 4 S.S. Wilks, "Mathematical Statistics," (John Wiley and Sons, Inc., New York, 1962).

W 5 M.T. Wasan, "Stochastic Approximation," Cambridge University Press, 1969.

W 6 T.J. Wagner, J.M. Pitt and B.F. Womack, "A comparison between pattern classification approaches," IEEE Trans. Information Theory (Correspondence), Vol. IT-13, 611-613, October, 1967.

Y 1 S.S. Yau and J.M. Schumpert, "Design of pattern classifiers with updating property using stochastic approximation techniques," IEEE Trans. on Computers, Vol. C-17, No. 9, September, 1968.

APPENDIX

APPENDIX A

CONVERGENCE PROOF

The convergence proof of (GA.4) can be divided into two possible situations.

Case 1. If the constraint on E is violated, the algorithm will be terminated since $\delta E[n] = 0$ for all n.

Case 2. Assume that the constraint on E holds. It must be shown that the algorithm converges. Before the proof of $\|D[n]\| \to 0$ as $n \to \infty$, two matrix identities will be proved.

$$\text{(a)} \quad \Psi^T D[n] = \Psi^T(\Psi A[n] - E[n])$$

$$= \Psi^T(\Psi\Psi^{\#}E[n] - E[n])$$

$$= (\Psi^T\Psi\Psi^{\#} - \Psi^T)\,E[n]$$

$$= (\Psi^T\Psi(\Psi^T\Psi)^{-1}\Psi^T - \Psi^T)\,E[n] = 0$$

$$\text{(b)} \quad \text{Trace}\,\{D[n]^T(\Psi\Psi^{\#} - I)^T(\Psi\Psi^{\#} - I)\,D[n]\}$$

$$= \text{Trace}\,\{D[n]^T[(\Psi\Psi^{\#})^T(\Psi\Psi^{\#}) - \Psi\Psi^{\#} - \Psi\Psi^{\#} + I]\,D[n]\}$$

$$= \text{Trace}\,\{D[n]^T[\Psi(\Psi^T\Psi)^{-1}\Psi^T\Psi(\Psi^T\Psi)^{-1}\Psi^T$$

$$- 2\Psi\Psi^{\#} + I]\,D[n]\}$$

$$= \text{Trace}\,\{D[n]^T(I - \Psi\Psi^{\#})D[n]\}$$

$$= \text{Trace}\,\{D[n]^T D[n]\}$$

$$- \text{Trace}\,\{D[n]^T\Psi\Psi^{\#}D[n]\}$$

$$= \|D[n]\|^2 - \text{Trace}\,\{(\Psi A[n] - E[n])^T\Psi\Psi^{\#}D[n]\}$$

$$= \|D[n]\|^2 - \text{Trace}\,\{(\Psi\Psi^{\#}E[n] - E[n])^T\Psi\Psi^{\#}D[n]\}$$

$$= \|D[n]\|^2 - \text{Trace}\ \{E[n]^T(\Psi\Psi^{\#}-I)^T\Psi(\Psi^T\Psi)^{-1} \times \Psi^T D[n]\}$$

$$\overset{\text{by a}}{=} \|D[n]\|^2$$

Define $V(D[n]) = \|D[n]\|^2$ a positive definite function.

$$\Delta V(D[n]) = V(D[n+1]) - V(D[n])$$

$$= V(\Psi A[n+1] - E[n+1]) - V(D[n])$$

$$= V(\Psi(A[n] + \Psi^{\#}\delta E[n]) - E[n+1]) - V(D[n])$$

$$= V(\Psi A[n] + \Psi\Psi^{\#}\rho D[n] - E[n] - \rho D[n]) - V(D[n])$$

$$= V(D[n] + \rho(\Psi\Psi^{\#} - I)D[n]) - V(D[n])$$

$$= \|D[n] + \rho(\Psi\Psi^{\#} - I)D[n]\|^2 - \|D[n]\|^2$$

$$= \text{Trace}\ \{[D[n] + \rho(\Psi\Psi^{\#} - I)D[n]]^T \times [D[n] + \rho(\Psi\Psi^{\#} - I)D[n]]\} - \|D[n]\|^2$$

$$= \text{Trace}\ \{D[n]^TD[n] + \rho D[n]^T(\Psi\Psi^{\#} - I)^TD[n] + \rho D[n]^T(\Psi\Psi^{\#} - I)D[n] + \rho^2 D[n]^T(\Psi\Psi^{\#}-I)^T(\Psi\Psi^{\#}-I)D[n]\} - \|D[n]\|^2$$

$$\overset{\text{by b}}{=} \|D[n]\|^2 + 2\rho\ \text{Trace}\ \{D[n]^T(\Psi\Psi^{\#}-I)D[n]\} + \rho^2\|D[n]\|^2 - \|D[n]\|^2$$

$$= 2\rho\ \text{Trace}\ \{D[n]^T(\Psi\Psi^{\#}-I)D[n]\} + \rho^2\|D[n]\|^2$$

$$= 2\rho\ \text{Trace}\ \{D[n]^T\Psi\Psi^{\#}D[n]\} - 2\rho\ \text{Trace}\ \{D[n]^TD[n]\} + \rho^2\|D[n]\|^2$$

$$\overset{\text{by a}}{=} -2\rho\|D[n]\|^2 + \rho^2\|D[n]\|^2$$

$$= -\|D[n]\|^2(2\rho - \rho^2)$$

For $0 < \rho \leq 2$, $2\rho - \rho^2 = \rho(2-\rho) \geq 0$, then $\Delta V(D[n]) \leq 0$ for all $D[n]$ and $\Delta V(D[n]) = 0$ if $D[n] = 0$.

By Lyapunov's stability theorem for discrete systems (K2),

$$\lim_{n\to\infty} V(D[n]) = \lim_{n\to\infty} \|D[n]\|^2 = 0 .$$