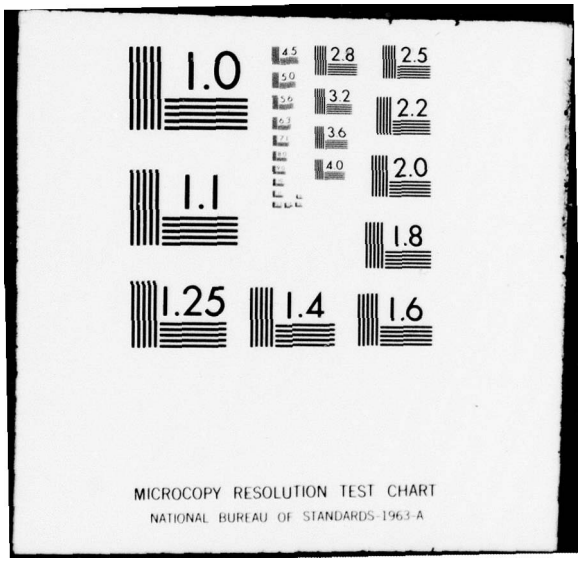
AD-A061 496 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 5/8
MATHEMATICAL PROGRAMMING APPLICATIONS IN PATTERN RECOGNITION.(U)
AUG 78 R H LEARY N00014-75-C-0267
UNCLASSIFIED SOL-78-14 NL

1 OF 2
AD A061496


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

12

Systems
Optimization
Laboratory

AD A061496

DDC FILE COPY

DDC
RECEIVED
NOV 24 1978
D

Department of Operations Research
Stanford University
Stanford, CA 94305

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

78 11 16 003

ACCESSION for
NTIS White Section X
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

LEVEL II

12

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
Stanford University
Stanford, California
94305

6

MATHEMATICAL PROGRAMMING APPLICATIONS IN PATTERN RECOGNITION.

10 by
Robert Hemstreet/Leary

9 TECHNICAL REPORT, SOL-78-14
August 1978

11 Aug 78

12 128p.

14 SOL-78-14

15

Research and reproduction of this report were partially supported by the Office of Naval Research Contract N00014-75-C-0267; the National Science Foundation Grant MCS76-81259 A01; and the Department of Energy Contract EY-76-S-03-0326 PA #18.

NSF-MCS76-81259

Reproduction in whole or in part is permitted for any purposes of the United States Government. This document has been approved for public release and sale; its distribution is unlimited.

DDC
RECEIVED
NOV 24 1978
D

408 765

AB

ACKNOWLEDGMENTS

I would like to express my appreciation and indebtedness to Professor George B. Dantzig for his guidance and patience. I am also grateful to Rosemarie Stampfel, who did the typing, Valerie Batt, who did the illustrations, and Gail Stein, who provided logistical support.

TABLE OF CONTENTS

CHAPTER 0

NOTATION

The following notations and conventions are used.

1. Lower case latin letters denote vectors, functions, and integers.
2. Upper case latin letters denote matrices and index sets.
3. Greek letters denote real numbers.
4. Script letters denote sets and classes.
5. The transpose of A is A'.
6. The i^{th} component of x is $(x)_i$.
7. No notational distinction will be made between row vectors and column vectors.
8. The inner product of the row vector w and the column vector x is denoted by $w \cdot x$.
9. The special vector e of dimension n, called the <u>unitary vector</u>, is defined by

$$(e)_i = 1 , \qquad i = 1,\ldots,n .$$

10. The following special functions are defined:

a) Positive part function:

$$\alpha^+ = \begin{cases} \alpha & \text{if } \alpha > 0 \\ 0 & \text{if } \alpha \leq 0 \end{cases}$$

b) Negative part function:

$$\alpha^- = \begin{cases} -\alpha & \text{if } \alpha < 0 \\ 0 & \text{if } \alpha \geq 0 \end{cases}$$

c) Sign function:

$$\operatorname{sgn}(\alpha) = \begin{cases} -1 & \text{if } \alpha < 0 \\ 0 & \text{if } \alpha = 0 \\ +1 & \text{if } \alpha > 0 \end{cases}$$

If any of these functions appears with a vector argument, the function applies to each component, i.e.

$$(f(x))_i = f((x)_i)$$

Similarly, the inequality $x > 0$ requires all components of x to be positive.

CHAPTER 1

INTRODUCTION

1.1. Pattern Recognition and Classification

Pattern recognition is concerned with the universal problem of identifying the "class" of an object from examination of its attributes. A major objective of pattern recognition is the development of machine implementable methods of classification. For simple applications such as optical reading of characters with a fixed type font, such methods offer great increases in speed and accuracy relative to human processing. For more difficult problems such as medical diagnosis or weather prediction, complex relationships in large quantities of multi-dimensional data may not immediately be apparent to casual observation. In such cases, algorithmic procedures implemented on a computer can often complement and extend human recognition capabilities.

The recognition process can be divided into two phases, feature extraction and classification. Feature extraction involves isolating the most relevant portions of the available data and representing them in a compact, useful form. A pattern is defined to be a finite dimensional vector $x \in \mathbb{R}^n$. Each component of the pattern is called a feature. Features are functions of observable data concerning the object to be classified. The feature extraction process consists of reducing points in a general measurement space to points in a finite dimensional pattern space.

For example, let the measurement space consist of continuous functions $f: \mathbb{R}^1 \to \mathbb{R}^1$ on the finite interval $[\alpha_1, \alpha_2]$. This case occurs in the analysis of electrocardiograms, electroencephalograms, various kinds of spectra, and generally in problems where the physical data consists of

continuous waveforms. A simple set of features can be generated by sampling the function on a uniform grid:

$$(x)_i = f(\alpha_1 + (i-1)\delta) , \qquad i = 1,\ldots,n$$

where

$$\delta = \frac{(\alpha_2 - \alpha_1)}{(n - 1)} .$$

Another alternative is to find an approximating function such as a polynomial that is defined by a finite set of parameters or coefficients which can then be used as the features. Clearly some feature sets will be better than others, but there are few if any general purpose feature extraction methods that yield good results for a wide variety of applications. Guess work, intuition, and experience with the specific problem are usually necessary to develop a good feature set.

Classification is concerned with determining decision procedures for assigning one of a finite number of class labels to a given pattern. The distinction between classification and feature extraction is not sharp, since classification itself may be a multi-stage process involving several transformations of the original pattern space.

Here we will be concerned with pattern classification procedures based on mathematical programming methods. Thus it is assumed that an initial set of features is given.

1.2. Discriminants and the Two-Class Problem

Let $x \in \mathbb{R}^n$ be a pattern that belongs to one of two possible classes, C_1 or C_2. One common form of classification rule decides

$$
\begin{aligned}
&x \in \mathcal{C}_1 \quad \text{if} \quad f(x) > 0 \\
&x \in \mathcal{C}_2 \quad \text{if} \quad f(x) < 0
\end{aligned}
\tag{1.2.1}
$$

where $f: \mathbb{R}^n \to \mathbb{R}^1$ is called a discriminant function. The case $f(x) = 0$ is considered indeterminant and an arbitrary decision may be made or the choice may be randomized with specified probabilities. Geometrically, the function f divides the pattern space into the disjoint regions

$$
\begin{aligned}
\mathcal{R}_1 &= \{x: f(x) > 0\} \\
\mathcal{R}_2 &= \{x: f(x) < 0\} \quad .
\end{aligned}
$$

For many classes of functions, the equation $f(x) = 0$ defines a surface that bounds these regions. In this case, $f(x)$ is said to separate $\mathcal{R}_1$ and $\mathcal{R}_2$.

There are several types of classification problems, each with its own solution philosophy. Two of these problems form the basis for much of the discussion here. The first, or template-matching problem, is characterized by two given finite sets of prototype pattern vectors x_j, one set for each class. Let

$$
\begin{aligned}
\mathcal{S}_1 &= \{x_1, \ \ldots \ , x_k\} \\
\mathcal{S}_2 &= \{x_{k+1}, \ \ldots \ , x_m\}
\end{aligned}
$$

be the prototype patterns for classes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Each observed pattern from a given class can be identified with one of the prototype patterns from that class, differing from the prototype by a

relatively small displacement vector d. The displacement vector can be thought of as a random error associated with the physical measurement process or as a statistical variation in the pattern population itself. Thus

$$C_i = \{x + d : x \in \mathcal{S}_i,\ d \in \mathcal{D}\}\ , \qquad i = 1,2 \tag{1.2.2}$$

where $\mathcal{D}$ is the set of possible displacement vectors.

The prototype sets $\mathcal{S}_1$, $\mathcal{S}_2$ are often called <u>design</u> or <u>training</u> sets. One general solution procedure for this problem involves assuming a parametric functional form $f(x;p)$ for the discriminant, where p is the parameter vector. The vector p is chosen so that $f(x_i;p) > 0$, $x_i \in \mathcal{S}_1$ and $f(x_i;p) < 0$, $x_i \in \mathcal{S}_2$, if possible, i.e. by solving the inequality system

$$\begin{aligned} f(x_i;p) &> 0\ , \qquad i = 1,\ldots,\ k \\ f(x_i;p) &< 0\ , \qquad i = k+1,\ \ldots\ ,\ m \end{aligned} \tag{1.2.3}$$

A feasible solution to (1.2.3) defines a discriminant that correctly classifies the design sets $\mathcal{S}_1$ and $\mathcal{S}_2$. If the system is feasible, the sets $\mathcal{S}_1$, $\mathcal{S}_2$ are said to be <u>separable</u> over the assumed parametric functional form. If the functional form $f(x;p)$ is continuous and the set of displacement vectors $\mathcal{D}$ is bounded by sufficiently small bounds, then the discriminant defined by this procedure will also separate the complete pattern classes C_1 and C_2.

Desirable properties of a discriminant function for the template-matching problem are errorless performance on the design sets $\mathcal{S}_1$, $\mathcal{S}_2$

and separation of $\mathcal{C}_1$ and $\mathcal{C}_2$ for the largest possible set of displacement vectors. Let $\mathcal{D} = \{d: \|d\| < \alpha\}$ for some vector norm $\|\cdot\|$. Then the template-matching problem for this vector norm is defined as

(1.2.4)
$$\max \alpha$$
$$\text{s.t. } f(x;p) > 0 , \qquad \forall\, x \in \mathcal{C}_1$$
$$f(x;p) < 0 , \qquad \forall\, x \in \mathcal{C}_2$$

where

$$\mathcal{C}_i = \{x + d: x \in \mathcal{S}_i,\ d \in \mathcal{D}\} , \qquad i = 1,2.$$

(See 1.2.1.)

A common choice of functional form is the linear discriminant

$$f(x) = w \cdot x - \theta$$

This case is quite general since any discriminant of the form

$$f(x) = \sum_{i=1}^{s} \alpha_i f_i(x) - \theta$$

is linear with respect to the transformed pattern $y \in \mathbb{R}^s$ defined by

$$(y)_i = f_i(x) , \qquad i = 1,\ldots,s .$$

Thus techniques developed for generation of linear discriminants are also applicable to all functions $f(x;p)$ that are linear in the parameter vector p, e.g. polynomials of all degrees in the components of x. The template-matching problem (1.2.4) for linear discriminants is discussed in the next chapter.

78 11 16 003

Template matching problems arise in relatively simple, well-defined contexts such as optical recognition of characters printed in a fixed type font, where pattern variation is very limited. More complex problems, such as medical diagnosis, often involve patterns that do not always fall into close groupings around prototypes. In this second kind of problem, observed patterns are considered as random samples from classes having different probability distributions. If the class distributions overlap, then an errorless classification scheme for the complete classes C_1 and C_2 is, of course, impossible. A discriminant is sought that minimizes some loss criterion such as the probability of misclassification.

This problem also involves two training sets $\mathcal{S}_1$, $\mathcal{S}_2$ consisting of examples of patterns from the respective classes C_1, C_2. A discriminant $f(x;p)$ is sought that performs well, although not necessarily perfectly, on the training sets. If these sets are large and well representative of their respective source distributions, then such a discriminant should perform well on these distributions. Some specific models and results for this type of problem are discussed in Chapter 4.

1.3. Outline of Presentation

Chapter 2 deals with classification problems for which linear discriminants can be found that separate the two design sets. Mathematical programming methods for determining these discriminants are discussed and reliability interpretations are made for a class of template-matching problems. An application to a set of adaptive pattern classification machines is given.

In Chapter 3 the least positive deviations solution concept for a possibly infeasible system of linear inequalities is defined. Connections with linear programming are established and a very efficient algorithm based on an unusual pivoting rule is developed for determining this solution. Application of the algorithm is extended to a sequence of problems of which the most general is the general linear programming problem.

In Chapter 4 this solution concept is applied to linearly inseparable classification problems. Large sample solution characterizations are obtained for design sets consisting of random samples from overlapping source distributions. Several alternative approaches to this problem are discussed and some numerical results utilizing the algorithm of Chapter 3 are presented.

Chapter 5 extends these methods to piecewise linear discriminants. A transformation of the pattern space is defined that renders any pair of finite, disjoint design sets separable by a convex piecewise linear function. An algorithm is presented that constructs such a function by solving a sequence of linear programs of a type directly suitable for application of the least positive deviations algorithm. Results for a sample problem are reported.

CHAPTER 2

LINEAR SEPARABLE CLASSIFICATION PROBLEMS

2.1. Linear Separability

Let $\mathcal{S}_1 = \{x_1, \ldots, x_k\}$, $\mathcal{S}_2 = \{x_{k+1}, \ldots, x_m\}$ be finite, disjoint, nonempty sets of n-dimensional patterns from classes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. These sets are defined to be linearly separable if there exists a linear discriminant $f(x) = w \cdot x - \theta$ such that

$$f(x) > 0 \qquad \forall\, x \in \mathcal{S}_1$$

$$f(x) < 0 \qquad \forall\, x \in \mathcal{S}_2.$$

The vector w is called the weight vector and the real number θ is called the threshold for reasons to be described in the next section. Geometrically, $\mathcal{S}_1$ and $\mathcal{S}_2$ are linearly separable if, as illustrated in Figure (2.1.1), there exists a separating hyperplane $w \cdot x = \theta$ such that all patterns in $\mathcal{S}_1$ lie in one half-space and all patterns in $\mathcal{S}_2$ lie in the other.

For each pattern $x \in \mathbb{R}^n$, a corresponding signed augmented pattern $a \in \mathbb{R}^{n+1}$ is defined by

$$a = \begin{cases} (x, -1), & \text{if } x \in \mathcal{S}_1 \\ (-x, +1), & \text{if } x \in \mathcal{S}_2 \end{cases} \tag{2.1.2}$$

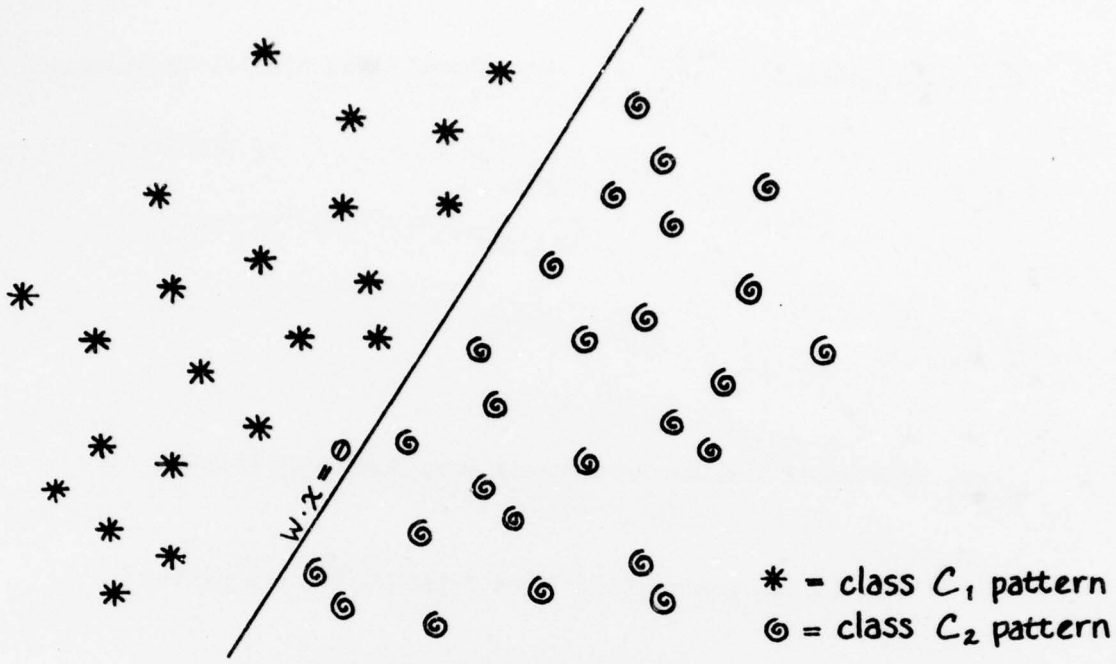


Figure (2.1.1). Linear Separable Pattern Sets.

The signed augmented pattern matrix $A \in \mathbb{R}^{m \times (n+1)}$ is defined by

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \left[\begin{array}{c|c} X_1 & -e_1 \\ \hline -X_2 & e_2 \end{array}\right] \tag{2.1.3}$$

where X_1 and X_2 are matrices whose rows are the patterns (row vectors) in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, and e_1 and e_2 are unitary column vectors.

PROPOSITION (2.1.4). $\mathcal{S}_1, \mathcal{S}_2$ are linear separable iff the inequality system

$$Au > 0 \tag{2.1.5}$$
$$u \in \mathbb{R}^{n+1}$$

is feasible.

Proof. The proof follows immediately from the identification $u = (w, \theta)$. □

Clearly the system (2.1.5) is feasible iff the system

$$Au \geq e \tag{2.1.6}$$
$$u = (w, \theta) \in \mathbb{R}^{n+1}$$

is feasible. System (2.1.6) will serve as the constraint set in several of the mathematical programming models discussed below. Application of the following version of the Farkas lemma provides a geometric criterion for linear separability.

LEMMA (Farkas). The inequality system

$$Au \geq b$$
$$u \in \mathbb{R}^{n}$$

is feasible iff the dual system

$$A'y = 0$$
$$b \cdot y > 0$$
$$y \geq 0, \quad y \in \mathbb{R}^{m}$$

is infeasible.

For

$$A = \begin{bmatrix} X_1 & \vdots & -e_1 \\ \cdots\cdots & \vdots & \cdots\cdots \\ -X_2 & \vdots & e_2 \end{bmatrix}$$

and $b = e$, the dual system is

$$\begin{aligned} X_1'y_1 - X_2'y_2 &= 0 \\ -e_1 \cdot y_1 + e_2 \cdot y_2 &= 0 \\ e_1 \cdot y_1 + e_2 \cdot y_2 &> 0 \\ (y_1, y_2) &\geq 0 \quad , \quad y_1 \in \mathbb{R}^k, \; y_2 \in \mathbb{R}^{m-k} \end{aligned} \tag{2.1.7}$$

Since the system is homogeneous and $e_1 \cdot y_1 = e_2 \cdot y_2 \neq 0$ any feasible solution $(\hat{y}_1, \hat{y}_2)$ can be scaled so that $e_1 \cdot \hat{y}_1 = e_2 \cdot \hat{y}_2 = 1$. Then $X_1'\hat{y}_1$ and $X_2'\hat{y}_2$ are points in the convex hulls of $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Thus the Farkas lemma stated geometrically says:

PROPOSITION (2.1.8). $\mathcal{S}_1$, $\mathcal{S}_2$ are linearly separable iff their respective convex hulls do not intersect.

2.2. Threshold Logic Units and Adaptive Machines

A device designed to implement a linear discriminant function is shown schematically in Figure (2.2.1). The device is called a threshold logic unit (TLU) and has aroused considerable interest as a simple mathematical model of a neuron (e.g. [1], [2], [3]). A TLU has n input terminals, one for each pattern component. Each pattern component $(x)_i$

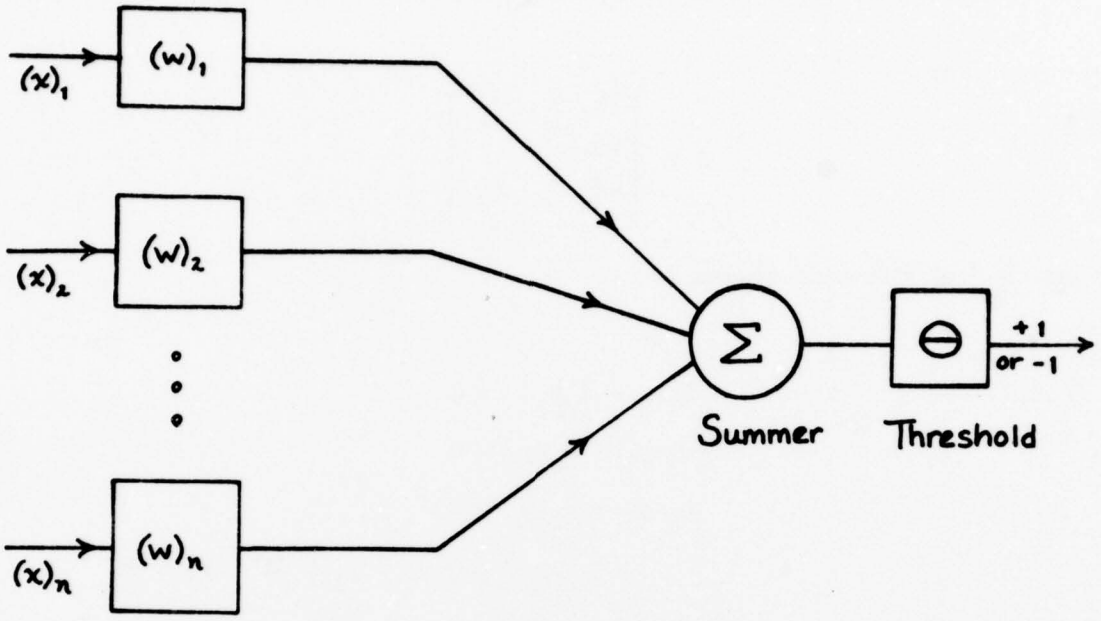


Figure (2.2.1). Threshold Logic Unit for Implementing the Discriminant $f(x) = w \cdot x - \theta$.

is multiplied by an adjustable internal weight $(w)_i$. The results are summed and compared to an adjustable threshold θ. An output of $+1$ is made if the sum equals or exceeds θ, otherwise the output is -1. (In the neuron model, the $+1$ output corresponds to the "firing" of a neuron in the presence of certain stimuli. The -1 output represents the normal, inactive state.)

The choice of the weight vector w and the threshold θ determine the patterns or stimuli that activate the TLU. Because the weights and threshold are adjustable, the TLU can be regarded as "trainable" and various adaptive algorithms have been devised for training. In particular,

error correction procedures have been investigated as training methods based on the following general scheme. Patterns are selected from $\mathcal{S}_1$ and $\mathcal{S}_2$ in some prescribed manner and presented to the TLU for classification. If a pattern is correctly classified, no corrective action is taken. If, however, the classification is incorrect, the weights and threshold are adjusted in a manner tending to correct the error.

A classic example is the Perceptron error correction procedure due to Rosenblatt [2]:

(2.2.2) Step 1. Set $u_1 = (w_1, \theta_1)$ to an arbitrary vector.
Set $k = 1$. Go to Step 2.

Step 2. Stop if u_k defines a separating hyperplane. Otherwise select any pattern $x \in \mathcal{S}_1 \cup \mathcal{S}_2$ which is incorrectly classified by u_k. Let a be the corresponding signed augmented pattern, so $u_k \cdot a \leq 0$. Go to Step 3.

Step 3. Set $u_{k+1} = u_k + a$. Increment k by 1 and go to Step 2.

It can be shown (Novikoff [4]) that if $\mathcal{S}_1$, $\mathcal{S}_2$ are linearly separable, then the algorithm converges in a finite number of steps to a separating u^*. Numerous variants to the procedure exist and a summary of error correction procedures for solution of the system $Au > 0$ is presented by Duda and Hart [5]. One major drawback to this class of methods is that they are generally ineffective in the linearly inseparable case in that no

determination of linear inseparability is made in a finite number of steps. This deficiency is corrected in the mathematical programming methods discussed later in this chapter.

More elaborate learning devices called Perceptrons can be constructed by assembling TLUs into layered networks such as that shown in Figure (2.2.3). Each TLU in the first, outer layer computes a binary function of the pattern vector. Subsequent inner layers perform Boolean operations on these binary functions. The innermost layer is a single TLU which makes the decision. The overall discriminant function implemented by such a network is piecewise linear, and with a sufficiently large number of TLUs, any two finite, disjoint pattern sets can be separated. Unfortunately, there is no known error-correction training algorithm analogous to (2.2.2) that is guaranteed to converge to a piecewise linear function capable of such a separation. Training is usually confined to the innermost TLU with the remaining weights being selected by heuristic or even random procedures. In Section (2.3) a linear programming procedure is presented for determining the weights and threshold in the inner layer that maximizes the reliability of a two-layer Perceptron when the outer layer TLUs are subject to failure. Also, in Chapter 5 mathematical programming methods are presented that determine a separating piecewise linear discriminant for general finite disjoint pattern sets.

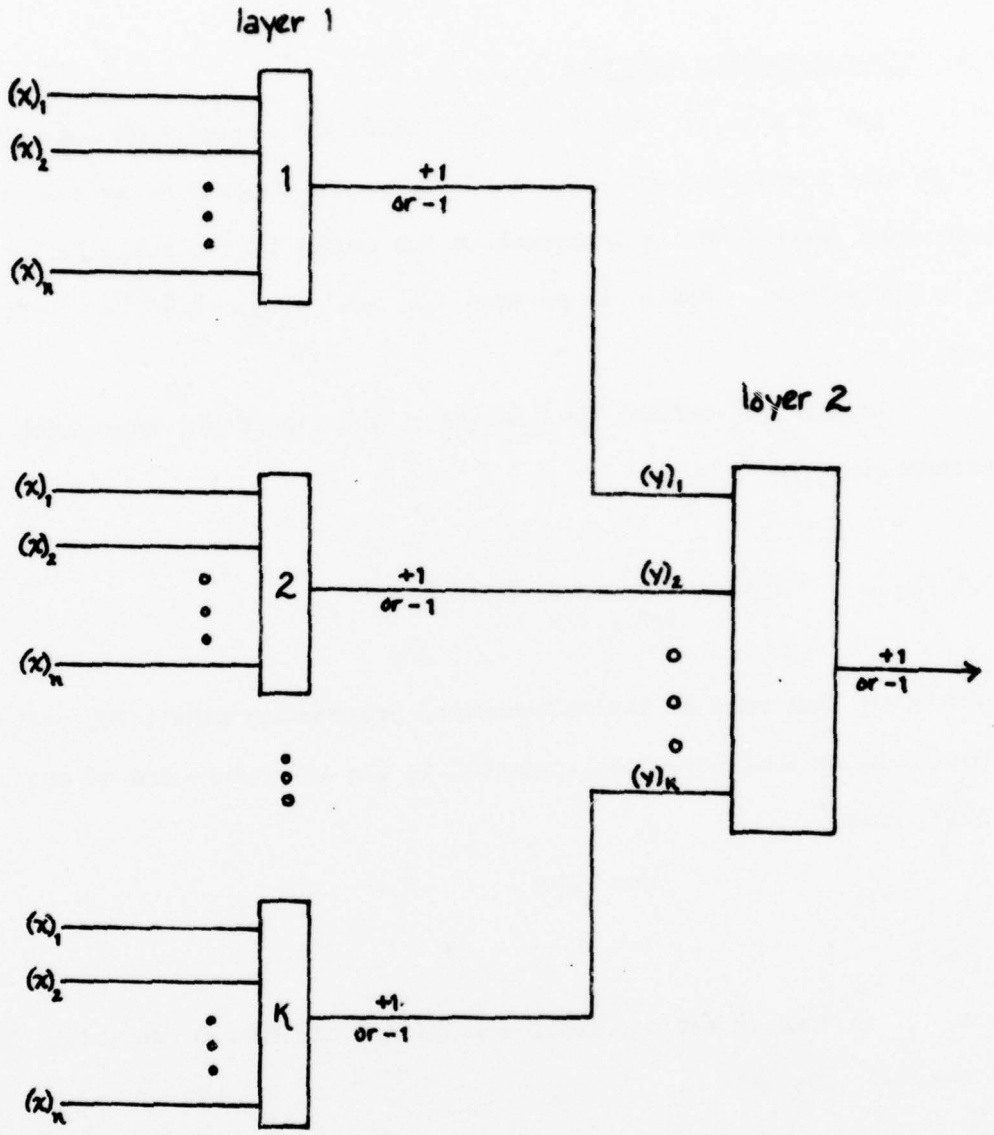


Figure (2.2.3). A Two-Layer Perceptron. The Output from the k First Layer TLUs Forms a k-dimensional Binary Input Vector y for the Second Layer TLU.

2.3. Maximum Quality Programs

Let $\hat{u} = (\hat{w}, \hat{\theta})$ solve $Au > 0$. Since the system is homogeneous, $\lambda\hat{u}$ is also a solution for any $\lambda > 0$; i.e. the underlying separating hyperplane $\lambda\hat{w}\cdot x = \lambda\hat{\theta}$ is invariant to the choice of the scale factor λ. It is convenient to scale $\hat{u}$ so that $\|\hat{w}\| = 1$, where $\|\cdot\|$ is a vector norm.

Grinold [6] defines the quality of the separating hyperplane corresponding to $\hat{u}$ as

$$Q(\hat{u}) = \min_{i=1,\ldots,m} \{(A\hat{u})_i\} \tag{2.3.1}$$

He observes that many of the mathematical programming models for pattern classification that have been suggested in the literature are of the general form

$$\begin{aligned} &\max \quad Q(u) \\ &\text{s.t.} \quad u \in U \end{aligned}$$

where U is some subset of feasible solutions to $Au > 0$ on which $Q(u)$ is bounded. The set

$$U = \{u : u = (w, \theta),\ Au > 0,\ \|w\| = 1\}$$

is a common choice that results in the mathematical program

$$\begin{aligned} &\max \quad \lambda \\ &\text{s.t.} \quad Au - \lambda e \geq 0 \\ &\qquad \|w\| \leq 1 \\ &\qquad u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned} \tag{2.3.2}$$

Let λ^* be the optimal objective value for (2.3.2). There are three possible cases.

Case 1: $\lambda^* = 0$. This corresponds to the optimal solution $u^* = 0$. It follows from Proposition (2.1.4) that $\mathcal{S}_1$, $\mathcal{S}_2$ are linearly inseparable.

Case 2: $0 < \lambda^* < \infty$. An optimal solution $u^* = (w^*, \theta^*)$ defines a separating hyperplane $w^* \cdot x^* = \theta^*$. The constraint $\|w^*\| \leq 1$ must be tight; otherwise $u = u^*/\|w^*\|$ is a better solution. Also, at least one of the constraints

$$Au^* - \lambda^* e \geq 0$$

is tight; otherwise $\lambda = \lambda^*$ may be increased while maintaining $u = u^*$. Hence $\lambda^* = Q(u^*)$.

Case 3: $\lambda^* = +\infty$. If $\mathcal{S}_1$ is empty, then $A = [-X_2 \vdots e_2]$ and $u = (w, \theta)$ is a feasible solution for all sufficiently large values of θ and any w satisfying $\|w\| \leq 1$. Hence λ is unbounded in this case and similarly in the case where $\mathcal{S}_2$ is empty and $A = [X_1 \vdots -e_1]$. If neither set is empty, then there exist at least two constraints of the form

$$w \cdot x_i - \theta \geq \lambda , \qquad x_i \in \mathcal{S}_1$$

$$-w \cdot x_j + \theta \geq \lambda , \qquad x_j \in \mathcal{S}_2$$

which imply

$$\lambda \leq \frac{1}{2} w(x_i - x_j)$$

Hence λ must be bounded since $\|w\| \leq 1$ and the sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are finite. Thus this case is eliminated by assuming neither sample set is empty.

These results are summarized in the following proposition:

PROPOSITION (2.3.3). Let $\mathcal{S}_1$, $\mathcal{S}_2$ be finite, non-empty pattern sets. Let $u^* = (w^*, \theta^*)$ be an optimal solution to (2.3.2) with objective value λ^*. Then $\mathcal{S}_1$, $\mathcal{S}_2$ are linearly separable iff $\lambda^* > 0$, and in the separable case $\|w^*\| = 1$ and $Q(u^*) = \lambda^*$.

If $\lambda^* > 0$, there is an equivalent form of (2.3.2):

$$\begin{aligned} &\min \|w\| \\ &\text{s.t.} \\ &\quad Au \geq e \\ &u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned} \tag{2.3.4}$$

The equivalence of (2.3.2) and (2.3.4) in the linearly separable case can be demonstrated by rewriting (2.3.4) as

$$\max \frac{1}{\|w\|} = \lambda$$

$$\text{s.t.} \quad A \frac{u}{\|w\|} - \lambda e \geq 0$$

$$u = (w, \theta) \in \mathbb{R}^{n+1}$$

Thus if $u^* = (w^*, \theta^*)$ solves (2.3.4), then $u^*/\|w^*\|$ solves (2.3.2) with $\max \lambda = 1/\|w^*\|$. Conversely, if $u^* = (w^*, \theta^*)$ solves (2.3.2)

with $\lambda^* > 0$, then u^*/λ^* solves (2.3.4) with $\min\|w\| = 1/\lambda^*$.

The programs (2.3.2) and (2.3.4) will be called the primary and alternative forms, respectively, of the maximum quality problem. Several different mathematical programming methods become applicable with the choice of the specific vector norm. Let

$$\|w\|_p = \left(\sum_{j=1}^{n} |(w)_j|^p \right)^{1/p}$$

denote the ℓ_p norm for $1 \leq p \leq \infty$, where

$$\|w\|_\infty = \max_{j=1,\ldots,n} \{|(w)_j|\}$$

The ℓ_∞, ℓ_1, and ℓ_2 norms are of particular interest since the maximum quality problem can be formulated as a linear program in the ℓ_∞ and ℓ_1 cases and as a quadratic program in the ℓ_2 case.

The ℓ_∞ norm case leads to the following linear program for the primary form of the maximum quality problem:

$$\begin{aligned} &\max \lambda \\ (2.3.5) \qquad &\text{s.t.} \\ &Au - \lambda e \geq 0 \\ &-1 \leq (w)_j \leq 1 \qquad j = 1,\ldots,n \\ &u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned}$$

This is a variation on a model originally proposed by Mangasarian [7].

System (2.3.5) has the following reliability interpretation for the two-layer Perceptron shown in Figure (2.2.3). In this TLU network, the first layer consists of k TLUs whose combined output forms a

transformed pattern $y \in \mathbb{R}^k$ for each input pattern $x \in \mathbb{R}^n$. The single second layer TLU then classifies y. The network is defined to be redundant if no final classification change results when the output of an arbitrary single TLU in the first layer changes from +1 to -1 or from -1 to +1. Thus a redundant Perceptron remains reliable with respect to any single TLU failure in the first layer.

Since the change induced by a failed TLU in the corresponding component of the transformed pattern y is of fixed magnitude, namely, 2, the discriminant function $f(y) = w \cdot y - \theta$ implemented by the inner layer TLU will not change sign if it is of sufficiently high quality. This is made explicit by the following proposition.

PROPOSITION (2.3.6). Let the set of transformed patterns be linearly separable by the hyperplane $w \cdot y = \theta$, where $\|w\|_\infty = 1$. Then the Perceptron is redundant if $Q(u) \equiv Q(w, \theta) > 2$.

Proof. Failure of the jth first layer TLU changes a transformed pattern y to y', where

$$|(y')_j - (y)_j| = 2 .$$

Let $f(y) = w \cdot y - \theta$. Then

$$|f(y') - f(y)| = 2|(w_j)| \leq 2, \qquad \text{since } \|w\|_\infty = 1$$

But $Q(w, \theta) > 2$ implies $|f(y)| > 2$. Hence $f(y)$ and $f(y')$ must be both either strictly positive or strictly negative and therefore no classification change occurs. □

If $Q(w,\theta) \leq 2$, the network may not be redundant. Since $|(w_j)| = 1$ for at least one component j, a failure in the corresponding TLU implies

$$|f(y') - f(y)| = 2$$

If, for example, $f(y) = -Q(w,\theta)$ and $f(y') = 2 + f(y)$, then

$$Q(w,\theta) \leq 2 \Rightarrow \begin{cases} f(y) < 0 \\ f(y') \geq 0 \end{cases}$$

Hence the inner layer TLU output changes from -1 to $+1$.

If $Q(w,\theta) > 2s$, where s is any positive integer, then by the argument used for Proposition (2.3.6), the Perceptron is redundant with respect to simultaneous failure of any s TLUs in the outer layer. Thus $Q(w,\theta)$ is an index of reliability in the sense described above and the maximum quality program (2.3.5) is a natural choice for determining the weights and threshold of the inner TLU when redundancy is a prime concentration.

A second formulation for solution of the linear separability problem is suggested by Ibaraki and Maroga [8]:

$$\begin{aligned} &\min \sum_{j=1}^{n} |(w)_j| \\ &\text{s.t. } Au \geq e \\ &u = (w,\theta) \in \mathbb{R}^{n+1} \end{aligned} \tag{2.3.7}$$

This is the alternative form of the maximum quality problem with the ℓ_1

norm. Since the objective function is convex, piecewise linear, and separable, (2.3.7) has the linear programming equivalent

$$\begin{aligned} &\min\ e\cdot w^{+} + e\cdot w^{-} \\ &\text{s.t. } Au^{+} - Au^{-} \geq e \\ &u^{+} \geq 0,\ u^{-} \geq 0,\quad u = (w,\theta) \in \mathbb{R}^{n+1} \end{aligned} \tag{2.3.8}$$

This program has m constraints in $2(n+1)$ non-negative variables. Typically m, the total number of patterns, is much larger than n, the pattern dimensionality. Thus it may be computationally advantageous to solve the dual

$$\begin{aligned} &\max\ e\cdot y \\ &\text{s.t. } \begin{pmatrix} -e \\ 0 \end{pmatrix} \leq A'y \leq \begin{pmatrix} e \\ 0 \end{pmatrix} \\ &y \geq 0,\quad y \in \mathbb{R}^{m} \end{aligned} \tag{2.3.9}$$

which has $(2n-1)$ constraints in m non-negative variables.

Let $w\cdot x = \theta$ be a separating hyperplane for the design sets $\mathcal{S}_1$, $\mathcal{S}_2$. In problems such as the template-matching model (1.2.4), it is desirable for the discriminant to generalize to additional patterns that differ from those in the design sets by small observation errors and noise terms. Ibaraki and Maroga define the <u>input tolerance</u> δ associated with $w\cdot x = \theta$ as the upper bound on the ℓ_∞ norm of displacement vectors d such that $x + d$ lies on the same side of the hyperplane as x, where $x \in \mathcal{S}_1 \cup \mathcal{S}_2$. Thus an observed pattern x' that differs from a design pattern x by a magnitude less than δ in each component will be classified into the same class as x. They show that the separating hyperplane defined by an optimal solution to (2.3.7) has the maximum input tolerance of all separating hyperplanes.

The alternative form of the maximum quality problem with the ℓ_2 norm was first investigated by Rosen [9] in the form of the quadratic program

$$\begin{aligned} &\min\ w\cdot w \\ \text{(2.3.10)} \qquad &\text{s.t.}\quad Au \geq e \\ &u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned}$$

This program has the following geometrical interpretation. Let $\hat{u} = (\hat{w}, \hat{\theta})$ be any feasible solution to the system $Au \geq e$. The hyperplanes $\hat{w}\cdot x = \hat{\theta}-1$, $\hat{w}\cdot x = \hat{\theta} + 1$ are parallel to the separating hyperplane $\hat{w}\cdot x = \hat{\theta}$ and bound a "dead zone" $\mathcal{D} = \{x: |\hat{w}\cdot\hat{x}-\hat{\theta}| < 1\}$ of width $2/\|w\|_2$ as shown in Figure (2.3.11). If the patterns in $\mathcal{S}_1$ and $\mathcal{S}_2$ suffer displacements,

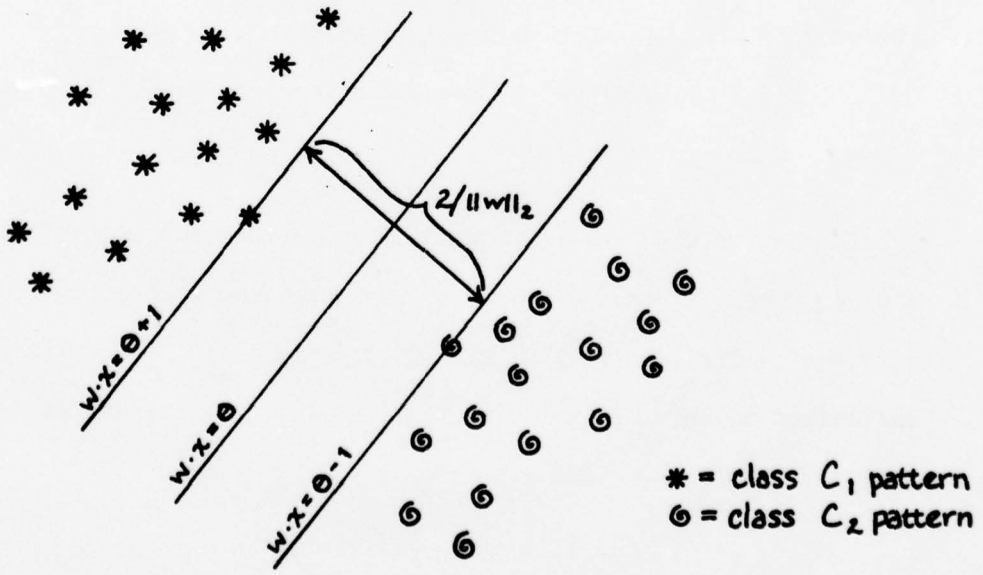


Figure (2.3.11). Pattern Sets Separated by an Empty "Dead Zone."

the hyperplane $\hat{w}\cdot x = \hat{\theta}$ will still separate the displaced sets as long as all displacements are of Euclidean distance less than $1/\|w\|_2$, i.e. half the width of the dead zone. Thus the optimal solution to (2.3.9) defines the separating hyperplane with dead zone of greatest width and hence highest tolerance to pattern displacements as measured by Euclidean distance.

The reliability results of this section are all examples of the following general principle. Let $u = (w,\theta)$ define a separating hyperplane $w\cdot x = \theta$ of quality $Q(u)$ for the pattern sets $\mathcal{S}_1$, $\mathcal{S}_2$. Let the pattern classes $\mathcal{C}_1$, $\mathcal{C}_2$ be defined for a given scalar value $\alpha > 0$ by

$$\mathcal{C}_i = \{x + d : x \in \mathcal{S}_i,\ \|d\|_q < \alpha\}$$

Let p, q be real numbers such that $1 \leq p \leq \infty$, $1 \leq q \leq \infty$, and $1/p + 1/q = 1$.

PROPOSITION (2.2.12). Let $w\cdot x = \theta$ separate $\mathcal{S}_1$, $\mathcal{S}_2$.

If $\alpha \leq Q(w,\theta)/\|w\|_p$, then the hyperplane $w\cdot x = \theta$ separates $\mathcal{C}_1$ and $\mathcal{C}_2$.

Proof. Let $f(x) = w\cdot x - \theta$. For a given displacement d and pattern x $x \in \mathcal{S}_1 \cup \mathcal{S}_2$, x and $x + d$ will have the same classification if $|f(x + d) - f(x)| < |f(x)|$. But $Q(u) = \min_{x \in \mathcal{S}_1 \cup \mathcal{S}_2} |f(x)|$, so it is sufficient to show $|f(x + d) - f(x)| < Q(u)$ for all displacements d such that $\|d\|_q < \alpha$. But

$$|f(x + d) - f(x)| = |w\cdot d|$$

and the result follows from the hypothesized upper bound on α and the

Hölder inequality

$$|w \cdot d| \leq \|w\|_p \cdot \|d\|_q \,. \qquad \square$$

For any w there exists a displacement d such that $|w \cdot d| = \|w\|_p \cdot \|d\|_q$ (Luenberger [10], p. 30). Thus the bound $Q(u)/\|w\|_p$ on the ℓ_q norm of allowable displacements is sharp, and the solution of the maximum quality problem (2.3.2) or (2.3.4) defines a separating hyperplane that maximizes the ℓ_q norm of allowable displacements. The Perceptron results ($p = \infty$, $q = 1$), the input tolerance in the Ibaraki and Maroga model ($p = 1$, $q = \infty$), and the dead zone width in the Rosen model ($p = 2$, $q = 2$) are specializations of the following corollary to Proposition (2.3.12).

COROLLARY (2.3.13). Let $\hat{w} \cdot x = \hat{\theta}$ be a separating hyperplane defined by an optimal solution $\hat{u} = (\hat{w}, \hat{\theta})$ to the maximum quality problem

$$\begin{aligned} &\max \lambda \\ &\text{s.t.} \quad Au - \lambda e \geq 0 \\ &\qquad \|w\|_p \leq 1 \\ &\qquad u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned}$$

Then $f(x) = w \cdot \hat{x} - \hat{\theta}$ solves the template matching problem (1.2.4) for the ℓ_q norm, i.e.

$$\begin{aligned} &\max \alpha \\ &\text{s.t.} \quad f(x;p) > 0 \quad \forall x \in \mathcal{C}_1 \equiv \{x + d : x \in \mathcal{S}_1,\ d \in \mathcal{D}\} \\ &\qquad f(x;p) < 0 \quad \forall x \in \mathcal{C}_2 \equiv \{x + d : x \in \mathcal{S}_2,\ d \in \mathcal{D}\} \end{aligned}$$

where $f(x;p) = f(x;w,\theta) = w \cdot x - \theta$ and $\mathcal{D} = \{d : \|d\|_q < \alpha\}$.

2.4. Extensions to the Inseparable Case

If the definition (2.3.1) of the quality $Q(u)$ of the hyperplane defined by $u = (w, \theta)$ is extended to non-separating hyperplanes, the quality of such hyperplanes is non-positive. In a linearly inseparable problem, a maximum quality hyperplane may provide a useful discriminant if the region of overlap between the convex hulls of $\mathcal{S}_1$ and $\mathcal{S}_2$ is relatively small. However, the maximum quality program (2.3.2) is no longer applicable since it produces the useless optimal solution $\hat{u} = 0$ in the inseparable case. This solution can be eliminated by bounding $\|w\|$ away from zero in the program

$$\begin{aligned} &\max \ \lambda \\ (2.4.1) \qquad &\text{s.t.} \ \ Au - \lambda e \geq 0 \\ &\|w\| \geq 1 \\ &u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned}$$

which is obtained from (2.3.2) by reversing the inequality defining the bound on $\|w\|$.

Let $u^* = (w^*, \theta^*)$ be an optimal solution to (2.4.1) with optimal objective value λ^*. There are three cases.

Case 1. $\lambda^* = +\infty$. This occurs when $\mathcal{S}_1$ and $\mathcal{S}_2$ are linearly separable. In this case there exists a solution $\hat{u} = (\hat{w}, \hat{\theta})$ with quality $Q(\hat{u}) > 0$ to the system

$$\begin{aligned} &Au > 0 \\ &\|w\| = 1 \\ &u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned}$$

Then for all $\alpha \geq 1$, $u = \alpha\hat{u}$ is feasible for (2.4.1) with corresponding objective value $\lambda = \alpha Q(\hat{u})$.

Case 2. $\lambda^* = 0$. This is a special case in which strict linear separability is impossible but there exists a $u^* = (w^*, \theta^*)$ such that

$$w^* \cdot x \geq \theta \qquad \forall x \in \mathcal{S}_1$$

$$w^* \cdot x \leq \theta \qquad \forall x \in \mathcal{S}_2$$

with at least one inequality being tight for a pattern from each sample set. Thus the convex hulls of $\mathcal{S}_1$ and $\mathcal{S}_2$ intersect only in a subset of the hyperplane $w^* \cdot x = \theta^*$.

Case 3. $\lambda^* < 0$. This is the linearly inseparable case of interest. The constraint $\|w\| \geq 1$ is tight; otherwise $u = u^*/\|w^*\|$ is a better solution. Similarly, at least one of the inequalities $Au^* - \lambda^* e \geq 0$ is tight; otherwise $\lambda = \lambda^*$ may be increased while maintaining $u = u^*$. Hence $\lambda^* = Q(u^*)$ and (2.4.1) defines a maximum quality hyperplane.

These cases are summarized in the following analog to Proposition (2.3.3).

PROPOSITION (2.4.2). Let $\mathcal{S}_1$, $\mathcal{S}_2$ be finite pattern sets. Then $\mathcal{S}_1$, $\mathcal{S}_2$ are linearly inseparable iff (2.4.1) has a bounded optimal objective value $\lambda^* \leq 0$ corresponding to an optimal solution $u^* = (w^*, \theta^*)$. If $\lambda^* < 0$, then $\|w^*\| = 1$ and $\lambda^* = Q(u^*)$.

If $\lambda^* < 0$, (2.4.1) has the alternative form

$$\begin{aligned} &\max \|w\| \\ \text{s.t.} \quad &Au \geq -e \\ &u = (w, \theta) \in \mathbb{R}^{n+1}. \end{aligned} \tag{2.4.3}$$

If $u^* = (w^*, \theta^*)$ solves (2.4.3), then $u^*/\|w^*\|$ solves (2.4.1) with $\max \lambda = -1/\|w^*\|$. Conversely, if $u^* = (w^*, \theta^*)$ solves (2.4.1) with $\lambda^* < 0$, then $u^*/-\lambda^*$ solves (2.4.3) with $\max\|w\| = -1/\lambda^*$.

A geometric interpretation of the maximum quality hyperplane produced by (2.4.3) in the linearly inseparable case with the ℓ_2 norm is shown in Figure (2.4.4). The inequality system $Au \geq -e$ is equivalent to

$$
\begin{aligned}
& w \cdot x_i - \theta \geq -1 \qquad \forall\, x_i \in \mathcal{J}_1 \\
& w \cdot x_i - \theta \leq 1 \qquad \forall\, x_i \in \mathcal{J}_2
\end{aligned}
\tag{2.4.5}
$$

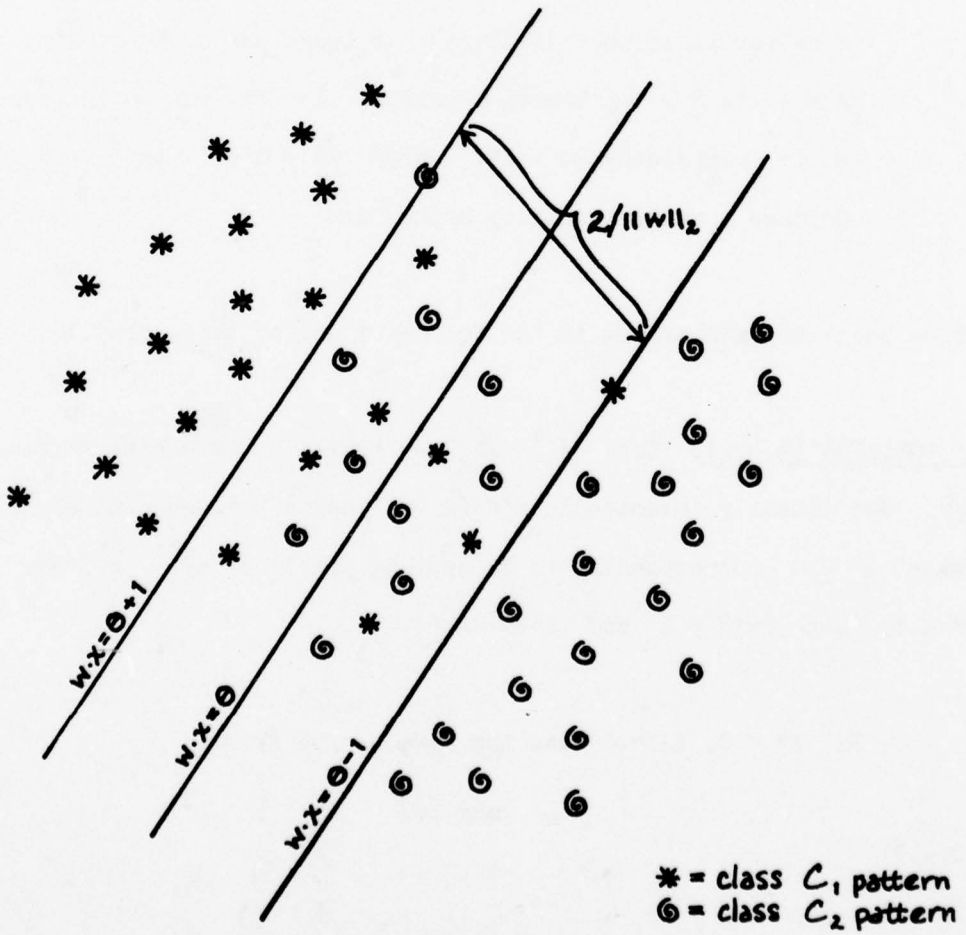


Figure (2.4.4). Linearly Inseparable Pattern Sets.

Thus all patterns in $\mathcal{S}_1$ lie in the non-negative half-space of the hyperplane $w \cdot x = \theta - 1$ and all patterns in $\mathcal{S}_2$ lie in the non-positive half-space of the parallel hyperplane $w \cdot x = \theta + 1$. Thus the pattern sets overlap in the zone

$$\mathcal{D} = \{x: |w \cdot x - \theta| \leq 1\} ,$$

while the patterns outside this zone all are classified correctly. The zone has width $2/\|w\|_2$, so the optimal solution to (2.4.3) is defined to be the one whose overlap zone is of minimum width.

The maximum quality hyperplane in the linearly inseparable case has several drawbacks. First, it is quite difficult to solve the programs (2.4.1) and (2.4.3) in general. In (2.4.1) the constant set is non-convex, while (2.4.3) requires the maximization of a convex function, so a Kuhn-Tucker point is not necessarily a global optimum. Second, the maximum quality hyperplane may be a very poor choice if there is significant overlap between the convex hulls of $\mathcal{S}_1$ and $\mathcal{S}_2$. The problem is illustrated by the following example, which shows that the maximum quality hyperplane places too much emphasis on the outlying or "maverick" patterns which are least representative of their own classes.

EXAMPLE (2.4.6).

Let $\mathcal{S}_1 = \{1, 2, \ldots, k, -(k+1)\}$, $\mathcal{S}_2 = \{-1, -2, \ldots, -k, (k+1)\}$ be sets of one-dimensional patterns. For $k \geq 2$, the linear discriminant with the lowest error rate is given by $f(x) = x - \theta$ for any $\theta \in (-1, 1)$. Such a discriminant misclassifies only the two outliers, namely, $-(k+1)$ in $\mathcal{S}_1$ and $(k+1)$ in $\mathcal{S}_2$. However, the maximum

quality hyperplane produced by (2.4.1) is quite different. Tightness of the constraints $\|w\| \geq 1$ at optimality implies that $w = +1$ or $w = -1$ for any ℓ_p norm. It is easily verified that the optimal solution to (2.4.1) is $(\hat{w},\hat{\theta}) = (-1,0)$ with $\hat{\lambda} = 1-k$. This corresponds to the discriminant $f(x) = -x$, which misclassifies all patterns in both sets except the two outliers. For large values of k this ranks among the worst choices of possible discriminants. □

The difficulty of computation and possible poor performance of the maximum quality hyperplane for linearly inseparable problems suggests the need for alternative procedures. Such procedures are the subject of Chapter 4. In particular, the linear program

$$\begin{aligned} &\min\ e\cdot s \\ &\text{s.t.}\quad Au + Is \geq e \\ &\qquad\quad s \geq 0 \\ &u \in \mathbb{R}^{n+1},\quad s \in \mathbb{R}^m \end{aligned} \tag{2.4.7}$$

is discussed. This program determines a separating hyperplane if one exists, but the solution does not necessarily have any of the desirable properties of a maximum quality discriminant. However, in the linearly inseparable case, (2.4.7) is much easier to solve than a maximum quality problem and places less emphasis on outlying patterns. For the example cited above, it is shown that the optimal solution to (2.4.7) yields the discriminant $f(x) = x$, which is in the set of lowest error rate discriminants.

CHAPTER 3

THE LEAST POSITIVE DEVIATIONS PROBLEM

3.1. Linear Inequalities

This chapter deals with the general linear inequality system

$$\begin{aligned} Ax &\geq b \\ x &\in \mathbb{R}^n \end{aligned} \tag{3.1.1}$$

where A is a $(m \times n)$ matrix with $m \geq n$ and $b \in \mathbb{R}^m$. The matrix A is assumed to be of full column rank n. Let a_i denote the ith row of A and β_i the ith component of b. Thus the ith inequality is $a_i \cdot x \geq \beta_i$.

A solution to (3.1.1), if one exists, can be found as the optimal solution to the Phase I linear program

$$\begin{aligned} &\min\ e \cdot s \\ \text{s.t.}\quad &Ax + Is \geq b \\ &s \geq 0 \\ &x \in \mathbb{R}^n,\quad s \in \mathbb{R}^m \end{aligned} \tag{3.1.2}$$

Problem (3.1.2) will be called the <u>least positive deviations</u> (LPD) problem corresponding to the tableau $[A:b]$. If $(\hat{x},\hat{s})$ is an optimal solution to (3.1.2), then $\hat{x}$ will be called a <u>LPD solution</u> to the inequality system (3.1.1). A LPD solution always exists since the LPD

linear program is feasible ($x = 0$, $s = b^+$ is a feasible solution) and the objective function is bounded below by zero on the constraint set. Clearly system (3.1.1) has a feasible solution iff the LPD solution is a feasible solution. In this case the optimal LPD objective value is equal to zero.

The least positive deviations terminology arises from the equivalence between the LPD linear program and the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^{m} (\beta_i - a_i \cdot x)^+ \tag{3.1.3}$$

$$= e \cdot (b - Ax)^+$$

where $(\beta_i - a_i \cdot x)^+ = \max(0, \beta_i - a_i \cdot x)$. If $(\hat{x}, \hat{s})$ is optimal for (3.1.2), then it is easy to show that $\hat{s} = (b - A\hat{x})^+$. Furthermore, for any $x \in \mathbb{R}^n$, $(x, (b - Ax)^+)$ is feasible for (3.1.2). Together these statements imply that $(\hat{x}, \hat{s})$ is optimal for (3.1.2) iff $\hat{x}$ is optimal for (3.1.3), where $\hat{s} = (b - A\hat{x})^+$.

The LPD linear program (3.1.2) can be written in the standard form

$$\min e \cdot s_1$$

$$\text{s.t.} \quad Ax_1 - Ax_2 + Is_1 - Is_2 = b \tag{3.1.4}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad s_1 \geq 0, \quad s_2 \geq 0$$

$$x_1 \in \mathbb{R}^n, \quad x_2 \in \mathbb{R}^n, \quad s_1 \in \mathbb{R}^m, \quad s_2 \in \mathbb{R}^m$$

This primal formulation has m constraints in $2(m + n)$ non-negative variables. The dual of (3.1.4) is

$$
\begin{aligned}
& \max\ b \cdot y \\
\text{(3.1.5)} \qquad \text{s.t.} \quad & A'y = 0 \\
& 0 \leq y \leq e \\
& y \in \mathbb{R}^m
\end{aligned}
$$

which has n constraints in m upper bounded non-negative variables. If the number of inequalities m in (3.1.1) is large relative to n, then the simplex method with upper bounds (Dantzig [11]) applied to the dual (3.1.5) would be computationally more convenient and probably more efficient than the standard simplex method applied to the primal (3.1.4).

When applied to (3.1.5), the simplex method with upper bounds terminates in a basic optimal solution $\hat{y}$ which has n basic variables $(\hat{y})_{i_1}, \ldots, (\hat{y})_{i_n}$ and $(m-n)$ non-basic variables. Each non-basic variable is either equal to its lower bound of zero or its upper bound of one, while the basic variables are equal to values lying in the interval $[0,1]$. The optimal basis defined by $\hat{y}$ consists of n column vectors $a_{i_1}, \ldots, a_{i_n}$ from A'. The simplex multiplier vector $\hat{x}$ corresponding to the optimal basis is the solution to the $(n \times n)$ linear equality system

$$
\begin{aligned}
& a_{i_1} \cdot x = \beta_1 \\
\text{(3.1.6)} \qquad & \vdots \\
& a_{i_n} \cdot x = \beta_n
\end{aligned}
$$

From the duality relationship between (3.1.4) and (3.1.5) it follows that $\hat{x}$ defines the optimal solution

$$
\begin{aligned}
x_1 &= \hat{x}^+ \\
x_2 &= \hat{x}^- \\
s_1 &= (b - A\hat{x})^+ \\
s_2 &= (b - A\hat{x})^-
\end{aligned}
\tag{3.1.7}
$$

to the primal (3.1.4) and hence $\hat{x}$ is a LPD solution to (3.1.1). The termination condition of the simplex method with upper bounds requires that the columns of A' price out as follows

$$
\begin{aligned}
& a_i \cdot \hat{x} = \beta_i && \text{if } (\hat{y})_i \text{ is basic} \\
& a_i \cdot \hat{x} \geq \beta_i && \text{if } (\hat{y})_i = 0 \text{ and is non-basic} \\
& a_i \cdot \hat{x} \leq \beta_i && \text{if } (\hat{y})_i = 1 \text{ and is non-basic}
\end{aligned}
\tag{3.1.8}
$$

If the primal solution (3.1.7) is non-degenerate, there are no basic slack variables equal to zero and the inequalities in (3.1.8) are strict. Thus if a non-basic optimal dual variable is at its lower bound the corresponding inequality in (3.1.1) is satisfied at $x = \hat{x}$, while if it is at its upper bound the inequality is violated (assuming a non-degenerate primal solution). If the dual variable is basic, the inequality is tight.

Thus the search for a LPD solution to (3.1.1) can be confined to simplex multiplier vectors associated with bases for the dual problem (3.1.5). The following terminology will be used to describe these vectors. A point $\hat{x} \in \mathbb{R}^n$ is defined to be a <u>basic</u> inequality solution to the inequality system $Ax \geq b$ if at least n of the m inequalities are tight at $x = \hat{x}$ and the corresponding row vectors $a_{i_1}, \ldots, a_{i_n}$ are linearly independent. If exactly n inequalities are tight, $\hat{x}$ is <u>non-degenerate</u>. For any linearly independent set of n row vectors $a_{i_1}, \ldots, a_{i_n}$, there is exactly one basic inequality solution $\hat{x}$ which

can be found by solving the linear equality system (3.1.6). Each basic inequality solution x to (3.1.1) defines the corresponding basic feasible solution to the primal linear program (3.1.4) given by

$$(3.1.9) \qquad (x_1, x_2, s_1, s_2) = (x^+, x^-, (b - Ax)^+, (b - Ax)^-)$$

Conversely, however, each basic feasible solution to (3.1.4) does not necessarily define a basic inequality solution to (3.1.1). For example, the basic feasible solution to (3.1.4)

$$(x_1, x_2, s_1, s_2) = (0, 0, b^+, b^-)$$

corresponds to the non-basic inequality solution $x = 0$ to (3.1.1).

Two basic inequality solutions x_1, x_2 are defined to be <u>adjacent</u> if the corresponding dual bases have exactly (n-1) column vectors of A' in common. Thus the simplex multiplier vectors x_k, x_{k+1} at successive iterations of the simplex method with upper bounds applied to the dual are adjacent basic inequality solutions to (3.1.1). Again, however, the basic feasible solutions defined by (3.1.9) for two adjacent basic inequality solutions x_1, x_2 to (3.1.1) are not necessarily adjacent basic feasible solutions to the primal problem (3.1.4). The reason is that the two sets of (m-n) basic slack variables can be completely different. The algorithm presented below for determining LPD solutions gains considerable computational efficiency by moving only between adjacent basic inequality solutions to (3.1.1), thus avoiding pivoting operations at intermediate basic feasible solutions to (3.1.4) where only slack variables are entering and leaving the basis.

If the simplex method with upper bounds is applied to the dual, the dual objective function $b \cdot y$ increases at each step (assuming non-degeneracy). However, it is not true in general that $f(x_{k+1}) < f(x_k)$ for the corresponding simplex multiplier vectors x_k, x_{k+1}, where $f(x) = e \cdot (b - Ax)^+$ is the LPD objective function. Thus intermediate multiplier vectors may be quite far from being optimal for the primal, and hence the dual problem must be iterated to completion to obtain a good (in this case, optimal) basic solution to the primal. From numerical experience on LPD problems of pattern recognition and control theory origin, it has been observed that the structure of the constraint set can be very complicated even in relatively small problems (e.g. $m \leq 1000$, $n \leq 11$) with consequent slow convergence of the simplex method with upper bounds.

In the next sections an algorithm for the LPD problem is presented that has proved to be very efficient on many numerical test problems, particularly on those in which the system $Ax \geq b$ is infeasible. The algorithm produces a finite sequence $\{x_k\}$ of basic inequality solutions to (3.1.1) that terminates in a LPD solution. Members of the sequence are shown to be obtainable as the simplex multiplier vectors corresponding to the path of bases produced by a modification to the usual pivot selection rules in the simplex method with upper bounds applied to the dual. Assuming non-degeneracy, the new pivot selection rules produce a decrease in the primal objective rather than an increase in the dual objective at each basis change (the upper and lower bounds on the dual variables do not enter into the calculation and are thus ignored).

3.2. The One-Dimensional LPD Problem

For the case $n = 1$, the inequality system (3.1.1) has the form

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} x \geq \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix} \tag{3.2.1}$$

where x is a scalar variable. It is easily seen that system (3.2.1) is feasible iff

$$\tau_1 \equiv \max_{\alpha_i > 0} \frac{\beta_i}{\alpha_i} \leq \min_{\alpha_i < 0} \frac{\beta_i}{\alpha_i} \equiv \tau_2$$

and any x in the interval $[\tau_1, \tau_2]$ is a feasible solution ($\tau_1 = -\infty$ if all α_i are non-positive; similarly $\tau_2 = +\infty$ if all α_i are non-negative). However, if the system (3.2.1) is infeasible, a more general approach is necessary to find a LPD solution. In place of the linear programming approach presented in the last section, a more direct solution technique for the LPD problem is discussed below. This method treats the problem in the unconstrained form

$$\min_{x \in \mathbb{R}^1} f(x) = \sum_{i=1}^{m} (\beta_i - \alpha_i x)^+ \tag{3.2.2}$$

Without loss of generality, it is assumed that $\alpha_i \neq 0$, $i = 1, \ldots, m$.

Let $f(x) = \sum_{i=1}^{m} f_i(x)$, where $f_i(x) = (\beta_i - \alpha_i x)^+$. A typical $f_i(x)$ for $\alpha_i > 0$ and $\alpha_i < 0$ is graphed in Figure (3.2.3). In either case the graph consists of two linear segments with a breakpoint at $x = \beta_i/\alpha_i$. At the breakpoint the slope increases by $|\alpha_i|$. Figure (3.2.4)

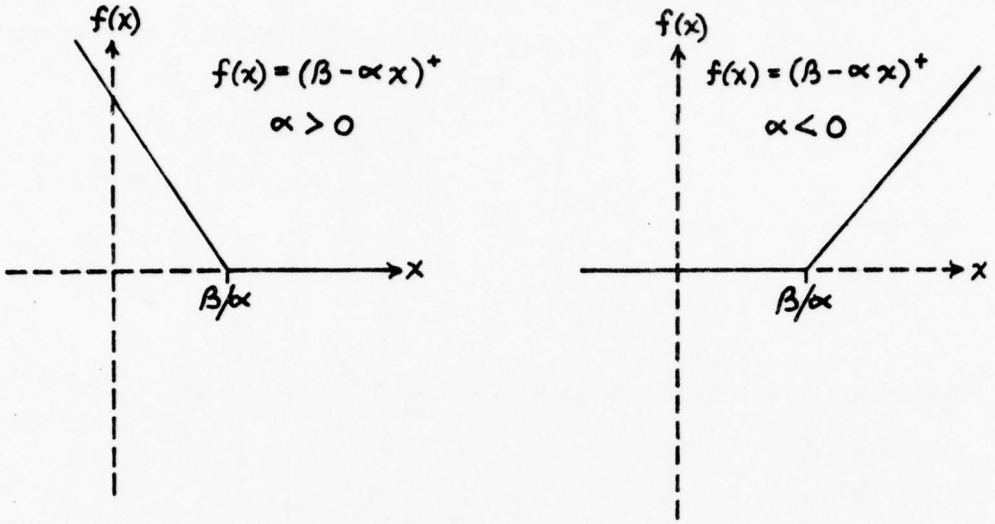


Figure (3.2.3). Typical One-dimensional Least Positive Deviation Functions.

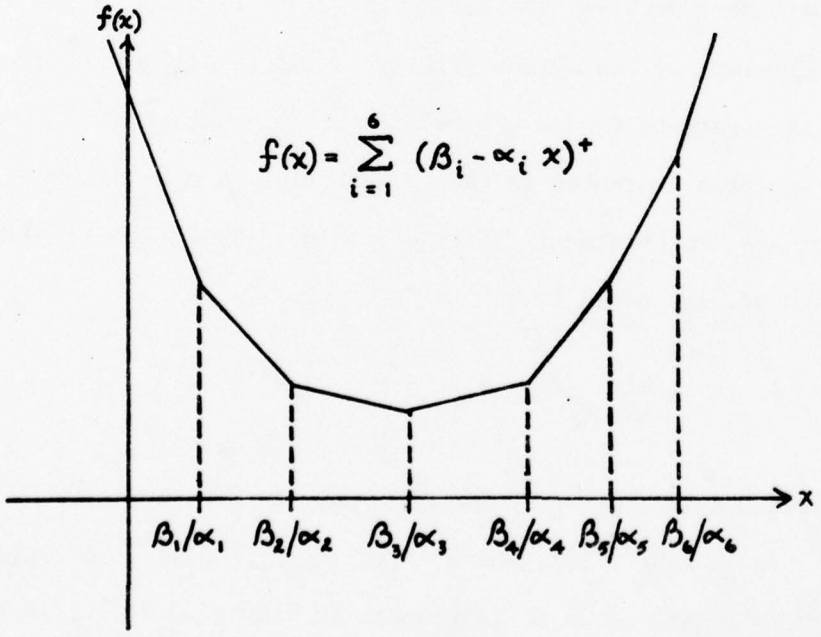


Figure (3.2.4). Sum of Six Positive Deviation Functions, with Minimum at $x = \beta_3/\alpha_3$.

illustrates the graph of the sum of six such functions. In general, assuming the points β_i/α_i, $i = 1,\ldots,m$ are distinct, the graph of $f(x)$ consists of $(m + 1)$ linear segments with breakpoints at each β_i/α_i. The right-hand derivative at each breakpoint β_i/α_i exceeds the left-hand derivative by $|\alpha_i|$. The extreme left hand infinite segment has slope equal to $-(\sum_{i=1}^{m} \alpha_i^+)$ and the extreme right-hand infinite segment has slope equal to $\sum_{i=1}^{m} \alpha_i^-$. If there is no flat (zero slope) segment, $f(x)$ has a unique minimum at the breakpoint where the right-hand derivative first becomes positive. If there is a flat segment, all points along this segment are minima.

From the observations above it follows that the right and left hand derivatives at any point $x = \hat{x}$ are given by the formulas

$$\left.\frac{df(x)}{dx^+}\right|_{x=\hat{x}} = -\sum_{i=1}^{m} \alpha_i^+ + \sum_{\{i:\beta_i/\alpha_i \leq \hat{x}\}} |\alpha_i|$$

(3.2.5)

$$\left.\frac{df(x)}{dx^-}\right|_{x=\hat{x}} = -\sum_{i=1}^{m} \alpha_i^+ + \sum_{\{i:\beta_i/\alpha_i < \hat{x}\}} |\alpha_i|$$

These formulas are the basis of the following solution procedure for the one-dimensional LPD problem (3.2.2)

PROCEDURE (3.2.6). One-Dimensional LPD Solution Procedure

1. Sort the m breakpoints β_i/α_i into ascending order. If there are repeated instances of any breakpoint, all such instances must be included in the ordered list. Reindex and let the index i now refer to the new order.

2. Let

$$\gamma_j = -\sum_{i=1}^{m} \alpha_i^+ + \sum_{i=1}^{j} |\alpha_i|, \qquad j = 1,\ldots,m$$

and let j^* be the smallest value of j for which $\gamma_j \geq 0$. Then the breakpoint $x^* = \beta_{j^*}/\alpha_{j^*}$ is optimal for (3.2.2).

Proof. By (3.2.5),

$$\gamma_j \leq \frac{df(x)}{dx^+}\bigg|_{x=\beta_j/\alpha_j}$$

with equality if only one inequality is tight at $x = \beta_j/\alpha_j$. Thus by definition of γ_j and j^*,

$$\frac{df(x)}{dx^-}\bigg|_{x=x^*} \leq 0 \leq \frac{df(x)}{dx^+}\bigg|_{x=x^*} \tag{3.2.7}$$

But (3.2.7) are precisely the necessary and sufficient conditions for a convex piecewise linear function of a scalar variable x to be minimized by $x = x^*$. □

The procedure is implemented by ordering the breakpoints and successively adding the slope changes $|\alpha_i|$ to the initial left-hand derivative. The procedure stops at an optimal breakpoint when this sum, and hence the right-hand derivative, first becomes non-negative. If the minimum is not unique, the solution that is produced is thus the smallest minimizing breakpoint.

As an alternative to (3.2.5), the derivatives at any point $\hat{x}$ can be calculated from the formulas

$$\left.\frac{df(x)}{dx^+}\right|_{x=\hat{x}} = \sum_{\{i:\alpha_i\hat{x}<\beta_i\}} -\alpha_i + \sum_{\{i:\alpha_i\hat{x}=\beta_i\}} \alpha_i^-$$

(3.2.8)

$$\left.\frac{df(x)}{dx^-}\right|_{x=\hat{x}} = \sum_{\{i:\alpha_i\hat{x}<\beta_i\}} -\alpha_i - \sum_{\{i:\alpha_i\hat{x}=\beta_i\}} \alpha_i^+$$

Rather than starting from the numerically smallest breakpoint as in Procedure (3.2.6), the search procedure can be initiated from an arbitrary breakpoint through the use of these formulas.

PROCEDURE (3.2.9). Modified One-Dimensional LPD Solution Procedure

1. Select an arbitrary breakpoint $\hat{x} = \beta_i/\alpha_i$ and calculate the left and right-hand derivatives at $\hat{x}$ from (3.2.8). If $\hat{x}$ is optimal by (3.2.7), stop. Otherwise go to step 2.
2. If $(df/dx^+)|_{x=\hat{x}} < 0$, sort the breakpoints that are strictly greater than $\hat{x}$ into ascending order and let the index i refer to this order. If $(df/dx^-)|_{x=\hat{x}} > 0$ sort the breakpoints that are strictly less than $\hat{x}$ into descending order indexed by i. In either case include all instances of repeated breakpoints.

Define

$$r_j = \left.\frac{df}{dx^+}\right|_{x=\hat{x}} - \text{sgn}\left(\left.\frac{df}{dx^+}\right|_{x=\hat{x}}\right) \cdot \sum_{i=1}^{j} |\alpha_j|$$

Let j^* be the smallest value of j such that

$$r_{j^*} \geq 0 \qquad \text{and} \qquad r_{j^*} - |\alpha_{j^*}| \leq 0$$

Then $x^* = \beta_{j^*}/\alpha_{j^*}$ is optimal for (3.2.2).

Proof. From (3.2.5) and (3.2.8), it follows that

$$r_j \leq \left.\frac{df}{dx^+}\right|_{x=\beta_j/\alpha_j}$$

$$r_j - |\alpha_j| \geq \left.\frac{df}{dx^-}\right|_{x=\beta_j/\alpha_j}$$

with equality in each case if only one inequality is tight at $x = \beta_j/\alpha_j$. Thus by definition of j^*, the optimality criterion (3.2.7) is satisifed at $x = x^*$. □

The procedure first determines on which side of the initial breakpoint $\hat{x}$ the minimum lies based on the algebraic sign of the right-hand derivative at $\hat{x}$. Successive breakpoints are then examined and the derivative updated until an optimum breakpoint satisfying (3.2.7) is found. Procedure (3.2.9) will be incorporated into an algorithm for solving the general n-dimensional LPD problem. It is used to solve one-dimensional problems of the form

$$\min_{\tau \in \mathbb{R}^1} f(x + \tau d)$$

where $x \in \mathbb{R}^n$ is a basic inequality solution to (3.1.1) and $d \in \mathbb{R}^n$ is a search direction.

Let $\hat{x} \in \mathbb{R}^n$ be a basic inequality solution to $Ax \geq b$ with $a_{i_1}, a_{i_2}, \ldots, a_{i_n}$ the linearly independent row vectors corresponding to n tight inequalities at $x = \hat{x}$. Define the $(n \times n)$ matrix

(3.2.10)
$$A_1 = \begin{pmatrix} a_{i_1} \\ a_{i_2} \\ \vdots \\ a_{i_n} \end{pmatrix}$$

Then the system $Ax \geq b$ can be rearranged and partitioned as

(3.2.11)
$$A_1 x \geq b_1$$
$$A_2 x \geq b_2$$

with $\hat{x} = A_1^{-1} b_1$. The following equations define a search direction $d^k \in \mathbb{R}^n$ such that all but the kth inequalities in $A_1 x \geq b_1$ remain tight for $x = \hat{x} + \tau d^k$, $\tau \neq 0$:

(3.2.12)
$$a_{i_j} \cdot d^k = 0 , \qquad j \neq k$$
$$a_{i_k} \cdot d^k = 1 .$$

Thus d^k is the kth column vector of A_1^{-1}. Improvement in the LPD objective function can be attempted by solving the one-dimensional problem

(3.2.13)
$$\min_{\tau \in \mathbb{R}^1} f(\hat{x} + \tau d^k)$$

This can be rewritten as the one-dimensional LPD problem

(3.2.14)
$$\min_{\tau \in \mathbb{R}^1} [e \cdot (\bar{b} - \bar{a}\tau)^+ + (-\tau)^+]$$

where
$$\bar{a} = A_2 d^k$$
$$\bar{b} = b_2 - A_2 \hat{x} .$$

Let $\tau^* = (\bar{b})_{i^*}/(\bar{a})_{i^*}$ be a minimizing breakpoint for (3.2.14).

PROPOSITION (3.2.15). If $\tau = 0$ is not optimal for problem (3.2.14), then $\hat{x} + \tau^* d^k$ and $\hat{x}$ are adjacent basic inequality solutions with $f(\hat{x} + \tau^* d^k) < f(\hat{x})$.

Proof. From the defining equations (3.2.12) for d^k, it follows that the n inequalities $A_1 x \geq b_1$, which are tight at $\tau = 0$, remain tight at $\tau = \tau^*$ except for the inequality corresponding to a_{i_k}. An additional inequality $a_{i^*} x \geq \beta_{i^*}$ in the system $A_2 x \geq b_2$ that was not tight at $\tau = 0$ becomes tight at $\tau = \tau^*$. The row vector a_{i^*} cannot be linearly dependent on $\{a_{i_1}, \ldots, a_{i_{k-1}}, a_{i_{k+1}}, \ldots, a_{i_n}\}$; otherwise $a_{i^*} \cdot d^k = 0$, implying $a_{i^*} \cdot \hat{x} = \beta_{i^*}$ and hence $(\bar{b})_{i^*} = 0$ which contradicts non-optimality at $\tau = 0$. Thus $\hat{x} + \tau^* d^k$ is a basic inequality solution corresponding to the dual basis $\{a_{i_1}, \ldots, a_{i_{k-1}}, a_{i^*}, a_{i_{k+1}}, \ldots, a_{i_n}\}$. Non-optimality at $\tau = 0$ implies $f(\hat{x} + \tau^* d^k) < f(\hat{x})$. □

Proposition (3.2.15) immediately suggests the algorithm for the LPD problem that is presented in the next section. Starting with an arbitrary basic inequality solution, the algorithm generates a sequence of improved adjacent basic inequality solutions by solving one-dimensional LPDs of the form (3.2.14). Computationally, the algorithm is shown to be implementable by changing the pivot selection rules of the simplex method with upper bounds as applied to the dual. As in the simplex method, the algorithm terminates in a finite number of steps with an optimal solution.

3.3. The ALPD Algorithm

This section presents the Accelerated Least Positive Deviations (ALPD) algorithm for determining a LPD solution to the system $Ax \geq b$. It will be assumed that all basic inequality solutions are non-degenerate. If this is not true, the vector b can be perturbed by

$$(b')_i = (b)_i + \epsilon^i$$

and the system $Ax \geq b'$ will be non-degenerate for all sufficiently small positive values of ϵ and the optimal dual basis for the perturbed system will also be optimal for the original system. The standard lexicographic schemes (Dantzig [11]) for the simplex method can be used intact.

Let

k = iteration number

x_k = basic solution at iteration k

A_k = $(n \times n)$ non-singular submatrix of A such that the corresponding inequalities are tight at x_k

d_k^i = ith column of A_k^{-1}

$f(x) = e \cdot (b - Ax)^+$.

ALGORITHM (3.3.1). ALPD

Step 0. Set $k = 1$. Let $A_1 x \geq b_1$ be any set of n inequalities such that A_1 is non-singular. Set $x_1 = A_1^{-1} b_1$. Go to Step 1.

Step 1. Determine the right and left-hand derivatives

$$\gamma_i = \left.\frac{df(x_k + \tau d_k^i)}{d\tau^+}\right|_{\tau=0} \qquad \delta_i = \left.\frac{df(x_k + \tau d_k^i)}{d\tau^-}\right|_{\tau=0}$$

for $i = 1, \ldots, n$.

Let $\lambda_i = \min(\gamma_i, -\delta_i)$ and $\lambda_{i*} = \min_{i=1,\ldots,n} \{\lambda_i\}$. If $\lambda_{i*} \geq 0$, go to Step 3. Otherwise, go to Step 2.

Step 2. Using (3.2.9) with initial breakpoint $\hat{\tau} = 0$, solve the one-dimensional LPD problem $\min_{\tau} f(x_k + \tau d_k^{i*})$. Let τ^* be the minimizing breakpoint. Set $x_{k+1} = x_k + \tau^* d_k^{i*}$ and form A_{k+1} by replacing the i*th row of A_k with the row of A corresponding to the breakpoint at τ^*. Increment k by 1 and go to Step 1.

Step 3. Stop. The final x_k is optimal.

PROPOSITION (3.3.2). Under the non-degeneracy assumption, the ALPD algorithm converges in a finite number of steps to an optimal basic solution.

Proof. Let x_k be an intermediate basic solution. Since x_k is not the final solution, by Step 1 and convexity of $f(x)$ the right and left-hand derivatives γ_{i*} and δ_{i*} are non-zero and have the same algebraic sign. Hence $\tau = 0$ cannot be optimal in Step 2 and therefore by Proposition (3.2.15) x_k and x_{k+1} are adjacent basic solutions with $f(x_{k+1}) < f(x_k)$. Thus cycling cannot occur. There are at most $\binom{m}{n}$ basic solutions, so the algorithm must be finite. It remains to be shown that the final basic

solution $\hat{x}$ is optimal. This will be done by demonstrating that the corresponding point

$$(x^+, x^-, s^+, s^-) = (\hat{x}^+, \hat{x}^-, (b - A\hat{x})^+, (b - A\hat{x})^-)$$

is optimal for the LPD linear program (3.1.4).

Let A_1 be the $n \times n$ submatrix of A corresponding to the basic tight inequalities at $x = \hat{x}$. Then the system $Ax \geq b$ can be written as

$$A_1 x \geq b_1$$

$$A_2 x \geq b_2$$

where A_2 is an $(m-n) \times n$ matrix corresponding to the remaining non-basic inequalities. The linear program (3.1.4) then takes the form

$$\min \quad e \cdot s_1^+ + e \cdot s_2^+$$

$$\text{s.t.} \quad A_1 x^+ - A_1 x^- + I s_1^+ - I s_1^- = b_1$$

$$A_2 x^+ - A_2 x^- + I s_2^+ - I s_2^- = b_2$$

$$x \in \mathbb{R}^n, \quad s_1 \in \mathbb{R}^n, \quad s_2 \in \mathbb{R}^{m-n}$$

Since A_1 is non-singular, this problem can be transformed by elementary row operations (Gaussian elimination) to yield

(3.3.3)
$$\min \quad e \cdot s_1^+ + e \cdot s_2^+$$

$$\text{s.t.} \quad I x^+ - I x^- + A_1^{-1} s_1^+ - A_1^{-1} s_1^- = A_1^{-1} b_1$$

$$I s_2^+ - I s_2^- - A_2 A_1^{-1} s_1^+ + A_2 A_1^{-1} s_1^- = b_2 - A_2 A_1^{-1} b_1$$

$$x \in \mathbb{R}^n, \quad s_1 \in \mathbb{R}^n, \quad s_2 \in \mathbb{R}^{m-n} .$$

The following basic feasible solution to (3.3.3) can be selected from (x^+, x^-, s^+, s^-):

$$\text{For } i = 1,\ldots,n \begin{cases} (x^+)_i \text{ is basic if } (A_1^{-1}b_1)_i > 0 \\ (x^-)_i \text{ is basic if } (A_1^{-1}b_1)_i < 0 \end{cases}$$

(3.3.4)

$$\text{For } j = 1,\ldots,m-n \begin{cases} s_2^+ \text{ is basic if } (b_2 - A_2A_1^{-1}b_1)_j > 0 \\ s_2^- \text{ is basic if } (b_2 - A_2A_1^{-1}b_1)_j < 0 \end{cases}$$

This solution corresponds to the basic solution $\hat{x} = A_1^{-1}b_1$ to (3.1.1). (3.3.4) is optimal if the reduced costs for all non-basic variables are non-negative.

Let (y_1,y_2), where $y_1 \in \mathbb{R}^n$, $y_2 \in \mathbb{R}^{m-n}$, be the simplex multiplier vector associated with the basic feasible solution (3.3.4). From the form of (3.3.3) it follows that $y_1 = 0$ and hence the reduced cost for all non-basic components of x^+ and x^- equals zero. Also from (3.3.3) it is seen that

(3.3.5)
$$\begin{aligned} (y_2)_j &= 1 \quad \text{if } (s_2^+)_j \text{ is basic} \\ (y_2)_j &= 0 \quad \text{if } (s_2^-)_j \text{ is basic} \end{aligned}$$

Thus the reduced cost for all the non-basic components of s_2^+ and s_2^- equals + 1. The reduced cost for $(s_1^+)_i$ is

(3.3.6) $$1 + y_2A_2d^i, \qquad i = 1,\ldots,n$$

where d^i is the ith column of A_1^{-1}. Similarly, the reduced cost for

$(s_1^-)_i$ is

$$-y_2 A_2 d^i , \qquad i = 1,\ldots, n. \tag{3.3.7}$$

Application of the formulas (3.2.8) to the function

$$f(\hat{x} + \tau d^i) = e\cdot(\bar{b} - \bar{a}\tau)^+ + (-\tau)^+ , \tag{3.3.8}$$

where $\bar{a} = A_2 d^i$ and $\bar{b} = b_2 - A_2 A_1^{-1} b_1$, yields

$$\left.\frac{df(\hat{x} + \tau d^i)}{d\tau^-}\right|_{\tau=0} = -\sum_{j=1}^{m-n} (A_2 d^i)_j^-$$

$$= -y_2 A_2 d^i$$

$$\equiv \gamma_i$$

and

$$\left.\frac{df(\hat{x} + \tau d^i)}{d\tau^-}\right|_{\tau=0} = -1 - \sum_{j=1}^{m-n} (A_2 d^i)_j$$

$$= -1 - y_2 A_2 d^i$$

$$\equiv \delta_i$$

Comparison with (3.3.6) and (3.3.7) reveals that γ_i and $-\delta_i$ are the reduced costs for $(s_1^-)_i$ and $(s_1^+)_i$, respectively.

But by the termination condition in Step 1,

$$\gamma_i \geq 0 , \qquad -\delta_i \geq 0 , \qquad i = 1,\ldots,n.$$

at $\hat{x}$, so $\hat{x}$ is optimal. □

The ALPD algorithm can be implemented using a pivotal procedure on tableaus. Define the tableau

$$T = \begin{bmatrix} A_1 & \vdots & b_1 \\ & \vdots & \\ A_2 & \vdots & b_2 \end{bmatrix}$$

corresponding to the basic solution $x = A_1^{-1}b_1$. By elementary column operations (equivalent to Gaussian elimination row operations on the transpose of T), T can be transformed into the canonical form

$$T_c = \begin{bmatrix} I & \vdots & 0 \\ A_2A_1^{-1} & \vdots & b_2 - A_2A_1^{-1}b_1 \end{bmatrix} \tag{3.3.9}$$

Let $\bar{a}^i$ be the ith column of $A_2A_1^{-1}$, $i = 1,\ldots,n$ and let $\bar{b} = b_2 - A_2A_1^{-1}b_1$. Then the right and left-hand derivatives γ_i, δ_i of the function $f_i(\tau) = e\cdot(\bar{b} - \bar{a}^i\tau)^+ + (-\tau)^+$ at $\tau = 0$ can be calculated from (3.2.8). The fastest rate of descent (minimum reduced cost for the primal problem) is

$$\lambda_{i*} = \min_{i=1,\ldots,n} \{\min(\gamma_i, -\delta_i)\} \tag{3.3.10}$$

If $\lambda_{i*} \geq 0$, then the current solution is optimal. Otherwise the i*th row of A_1 will leave the dual basis. It will be replaced by the j*th row of A_2, where

$$(\bar{b})_{j*}/(\bar{a}^{i*})_{j*} \tag{3.3.11}$$

is the minimizing breakpoint of $f_{i*}(\tau)$.

This is accomplished by executing a standard simplex method pivot on the transpose of T_c, using the $(i^*, j^* + n)$ element of T_c' as the pivot element. The new tableau, after rearrangement, will be in the canonical form corresponding to the new dual basis.

The pivot operation used to move from one basic solution to an adjacent basic solution is thus the same as that used by the simplex method with upper bounds applied to the dual (3.1.5) in exchanging one basic column of A' for another. Thus the ALPD algorithm can be implemented simply by changing the pivot selection rule in standard simplex method software and ignoring the upper bounds on the dual variables. The new pivot rule selects the column that leaves the dual basis according to the minimum reduced cost rule (3.3.10). The entering column is selected as the one corresponding to the breakpoint that minimizes the LPD objective function (3.3.9). Each iteration then results in a decreased primal objective rather than increased dual objective.

The relative efficiencies of the ALPD and simplex method pivot selection rules can be compared directly by counting the number of basis changes (pivot operations) required to reach optimality from a given starting basic. The ALPD algorithm has been coded in FORTRAN and applied to numerous small (typically $m \leq 1000$, $n \leq 11$) LPD problems arising from pattern classification models. These problems generally have a totally dense A with $b > 0$. Comparative runs with the simplex method with upper bounds have been made with the following general results. In cases where the inequality system $Ax \geq b$ is feasible, the two pivot rules require approximately the same number of pivots. However, in cases where the system is infeasible, the number of simplex method pivots grows rapidly with the extent of the infeasibility, i.e. the number of

inequalities violated by the optimal solution. The number of ALPD pivots appears insensitive to this factor. In many infeasible cases the number of simplex pivots exceeded the number of ALPD pivots by factors of several hundred. Detailed results of a series of systematic comparison trials are presented in the next chapter.

3.4. Initializing the Algorithm

The choice of the initial basic solution is arbitrary. However, the following procedure produces an initial basic solution by constructing a sequence $\{x_0, x_1, \ldots, x_n\}$ of points such that

$$f(x_k) \leq f(x_{k-1}), \qquad k = 1,\ldots,n$$

The final point x_n is the desired initial basic solution. Thus a considerable amount of improvement in the objective function may be achieved in the initiation sequence.

Let

k = iteration number

$\mathcal{B}_k$ = partial set of vectors in the dual basis at iteration k.

PROCEDURE (3.4.1). ALPD Initialization.

Step 1. Set $\mathcal{B}_0 = \emptyset$, $x_0 = 0$.

Choose an arbitrary direction $d_0 \neq 0$ (the unit vector $d_0 = (1,0,\ldots, 0)$ is convenient); set $k = 1$ and go to Step 2.

Step 2. Solve the one-dimensional LPD problem

$$\min_{\tau} f(x_{k-1} + \tau\, d_{k-1})$$

Let a^k be the row vector corresponding to the inequality that becomes tight at the optimizing $\tau = \tau^*$. Set

$$\mathcal{B}_k = \mathcal{B}_{k-1} \cup \{a^k\} \qquad x_k = x_{k-1} + \tau^* d_{k-1}$$

If $k = n$, go to Step 4. Otherwise go to Step 3.

Step 3. Determine a new direction $d_k \neq 0$ such that

$$d_k \cdot a^i = 0 , \qquad i = 1, \ldots, k,$$

(The Gram-Schmidt orthogonalization procedure can be used.)

Increment k by one and go to Step 2.

Step 4. Stop. $\mathcal{B}_n$ is the initial dual basis and x_n is the initial basic solution.

After Step 2 of iteration k, the k inequalities corresponding to $a^1, \ldots, a^k$ are tight at x_k. Each new search direction is generated in such a way that these inequalities remain tight at x_{k+1}, where a new inequality becomes tight.

3.5. Extensions of the LPD Problem

In this section a sequence of increasingly general linear programs are shown to be reducible to equivalent LPD linear programs. Ultimately the applicability of the ALPD algorithm to the general linear programming problem is demonstrated.

The weighted LPD problem with tableau [A:b] and weight vector $w > 0$ is defined as the linear program

$$\begin{aligned} &\min\ w\cdot s \\ \text{(3.5.1)} \qquad &\text{s.t.}\quad Ax + Is \geq b \\ &\qquad\quad s \geq 0 \\ &\qquad\quad x \in \mathbb{R}^n,\ \ s \in \mathbb{R}^m \end{aligned}$$

This is the immediate generalization of the standard LPD problem (3.1.2) obtained by replacing the LPD objective $e\cdot s$ with the weighted LPD objective $w\cdot s$. The equivalent unconstrained problem is

$$\text{(3.5.2)} \qquad \min_{x\in \mathbb{R}^n}\ f(x) = w\cdot(b - Ax)^+$$

Since $w > 0$, $w\cdot(b - Ax)^+ = e\cdot(Wb - WAx)^+$ where W is the $m \times m$ diagonal matrix defined by $W_{ii} = (w)_i$, $i = 1,\ldots,m$. Thus (3.5.1) is equivalent to a standard LPD problem with tableau [WA:Wb]. In application of the ALPD algorithm to (3.5.1), either of the tableaus [A:b] or [WA:Wb] may be used for pivoting since both have the same set of basic solutions. However, the pivot selections are governed by the derivatives of the LPD objective function $f(x) = e\cdot(Wb - WAx)^+$.

The weighted LPD problem can be further generalized by allowing penalties on both positive and negative deviations. Let $w \in \mathbb{R}^m$, $z \in \mathbb{R}^m$ be non-negative vectors such that $w + z > 0$. The weighted least deviations problem with tableau [A:b] and weight vectors w and z is defined by the linear program

$$\begin{aligned} &\min\ w\cdot s_1 + z\cdot s_2 \\ \text{(3.5.3)} \qquad &\text{s.t.}\quad Ax + Is_1 - Is_2 = b \\ &\qquad s_1 \geq 0\,, \quad s_2 \geq 0 \\ &\qquad x \in \mathbb{R}^n,\ \ s_1 \in \mathbb{R}^m,\ \ s_2 \in \mathbb{R}^m \end{aligned}$$

It is easily seen that in an optimal solution $(\hat{x}, \hat{s}_1, \hat{s}_2)$ to (3.5.3) the relations $\hat{s}_1 = (b - A\hat{x})^+$ and $\hat{s}_2 = (b - A\hat{x})^-$ must hold and hence (3.5.3) is equivalent to the unconstrained problem

$$\text{(3.5.4)} \qquad \min_{x \in \mathbb{R}^n} f(x) = w\cdot(b - Ax)^+ + z\cdot(b - Ax)^-$$

Since $(b - Ax)^- = (-b + Ax)^+$, the weighted least deviations problem (3.5.3) can be reformulated as a weighted LPD problem. In particular, if $w > 0$ and $z > 0$ the tableau and weight vector for this weighted LPD problem are

$$\text{(3.5.5)} \qquad \begin{bmatrix} A & b \\ \vdots & \vdots \\ -A & -b \end{bmatrix} \quad \text{and} \quad [w, z]$$

respectively. In application of the ALPD algorithm to (3.5.3) it is sufficient to pivot on the partial tableau [A:b] since the pivoting operation preserves the opposite sign relationship between the upper and lower halves of the full tableau. Again, however, the pivot selections are determined by the derivatives of (3.5.4).

The dual of (3.5.3) is

$$\begin{aligned} &\max\ b\cdot y \\ \text{(3.5.6)} \qquad &\text{s.t.}\quad A'y = 0 \\ &\qquad -z \leq y \leq w \\ &\qquad y \in \mathbb{R}^m \end{aligned}$$

which differs from the dual (3.1.5) of the standard LPD problem only in the generalization of the lower and upper bounds on the dual variables.

Example (3.5.6).

The general linear approximation problem with ℓ_p norm criterion is

$$\min_{x \in \mathbb{R}^n} f(x) = \|Ax - b\|_p \tag{3.5.7}$$

The vector $b \in \mathbb{R}^m$ is approximated by a linear combination of the columns of the $m \times n$ matrix A, with the best approximation defined as that which minimizes the ℓ_p norm of the residual vector. Problems of this type arise in linear regression analysis and function approximation ('curve fitting'). The choice $p = 2$, equivalent to the usual least squares criterion in regression analysis, is the simplest case both analytically and computationally, since an explicit solution $x = (A'A)^{-1} A'b$ exists whenever $A'A$ is non-singular. Also, in the general linear statistical model with the usual Gaussian error distribution assumption, the maximum likelihood estimate of the coefficient vector is a solution to a problem of type (3.5.7) with ℓ_2 norm. However, it was first suggested by Edgeworth [12] that the ℓ_1 criterion of minimizing the sum of the absolute values rather than the sum of the squares of the deviations may be more suitable when the deviations are large and erratic. For example, if the error distribution is given by the double exponential distribution with probability density

$$f(\epsilon) = (2\sigma)^{-1} e^{-|\epsilon|/\sigma}, \qquad -\infty < \epsilon < \infty$$

then the maximum likelihood estimate is a solution to a linear approximation problem with ℓ_1 norm criterion (Draper and Smith [13]). This distribution has a much more slowly decaying tail than a normal distribution with the same variance σ^2 and hence is more likely to produce the kind of deviation pattern mentioned above. Before the availability of linear programming techniques, however, the computational difficulties presented by the ℓ_1 criterion limited application of early solution methods (e.g. Rhodes [14] and Singleton [15]) to low dimensional problems, typically $n \leq 3$. The first linear programming formulation of this problem is due to Charnes and Cooper [16], who present the least weighted deviations program (3.5.3) with $w = e$ and $z = e$. The computational advantage of the dual and the applicability of the simplex method with upper bounds is noted by Wagner [17]. The ALPD algorithm given here is a generalization of a special purpose algorithm for the ℓ_1 norm problem presented without proof by Davies [18]. □

The final extension of the LPD problem considered here is the constrained weighted LPD problem obtained by adding the $p \times n$ inequality system

$$A_2 x \geq b_2 \qquad x \in \mathbb{R}^n \tag{3.5.8}$$

to the constraint set of the weighted LPD problem (3.5.1). (A constrained weighted least deviations problem can similarly be defined by adding (3.5.8) to the constraint set of the weighted least deviations problem (3.5.3). As shown above, the weighted least deviations problem can be reformulated as a weighted LPD problem, so the discussion below also applies to this case.) The general form of the constrained problem is thus

$$\begin{aligned} & \min\ w\cdot s \\ (3.5.9) \qquad \text{s.t.} \quad & A_1x + Is \geq b_1 \\ & A_2x \qquad \geq b_2 \\ & s \geq 0 \\ & x \in \mathbb{R}^n, \quad s \in \mathbb{R}^m \end{aligned}$$

The $(m + p) \times n$ matrix

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix},$$

where $m + p \geq n$, is assumed to be of full column rank n.

Problem (3.5.9) does not have a direct weighted LPD equivalent. However, if the inequality system (3.5.8) is feasible, the weighted LPD problem

$$\begin{aligned} & \min\ w\cdot s_1 + \lambda(e\cdot s_2) \\ (3.5.10) \qquad \text{s.t.} \quad & A_1x + Is_1 \qquad \geq b_1 \\ & A_2x \qquad + Is_2 \geq b_2 \\ & s_1 \geq 0, \quad s_2 \geq 0 \\ & x \in \mathbb{R}^n, \quad s_1 \in \mathbb{R}^m, \quad s_2 \in \mathbb{R}^p \end{aligned}$$

will be shown to have the same set of optimal basic solutions as (3.5.9) for sufficiently large values of the scalar weighting factor λ.

PROPOSITION (3.5.11). If the added constraints (3.5.8) are feasible, then the constrained weighted LPD problem (3.5.9) has an optimal solution.

Proof. If $\hat{x}$ is feasible for (3.5.8), then $x = \hat{x}$, $s = (b - A\hat{x})^+$ is feasible for (3.5.9). The objective function of (3.5.9) is bounded below by zero, so an optimal solution exists. □

LEMMA (3.5.12). Let x^* be an optimal basic solution to (3.5.10) for some $\lambda > 0$ and let $s_1^* = (b_1 - Ax^*)^+$. If $A_2x^* \geq b_2$, then (x^*, s^*) is optimal for (3.5.9).

Proof. If $A_2x^* \geq b_2$, then there exists an optimal solution $(\hat{x}, \hat{s})$ to (3.5.9) by Proposition (3.5.11). Since (x^*, s_1^*) is feasible for (3.5.9), $w \cdot s_1^* \geq w \cdot \hat{s}$. But $(x, s_1, s_2) = (\hat{x}, \hat{s}, 0)$ is feasible for (3.5.10) whereas $(x^*, s_1^*, 0)$ is optimal, so $w_1 s_1^* \leq w \cdot \hat{s}$. Thus $w \cdot s_1^* = w \cdot \hat{s}_1$ and (x^*, s_1^*) is optimal for (3.5.9). □

PROPOSITION (3.5.13). If the added constraints (3.5.8) are feasible, then there exists a number $\bar{\lambda} \geq 0$ such that any optimal basic solution x^* to (3.5.10) for $\lambda > \bar{\lambda}$ defines an optimal solution $x = x^*$, $s = (b_1 - Ax^*)^+$ to (3.5.9).

Proof. Let $\mathcal{B} = \{x_1, \ldots, x_k\}$ be the set of basic solutions to (3.5.10) that are infeasible for the system $A_2x \geq b_2$. $\mathcal{B}$ is a finite, possibly empty set. If $\mathcal{B}$ is empty, let $\bar{\lambda} = 0$ and the result follows from the lemma. If $\mathcal{B}$ is non-empty, define

$$\delta = \min_{x_i \in \mathcal{B}} e \cdot [b_2 - A_2x_i]^+$$

and let $\bar{\lambda} = (w \cdot \hat{s})/\delta$ where $(\hat{x}, \hat{s})$ is any optimal solution to (3.5.9). Thus if $\lambda > \bar{\lambda}$,

$$\lambda(e \cdot [b_2 - Ax_i]^+) > w \cdot \hat{s} \qquad \forall\, x_i \in \mathcal{B}$$

Hence no member of $\mathcal{B}$ can be optimal for (3.5.10) when $\lambda > \bar{\lambda}$ since $(x, s_1, s_2) = (\hat{x}, \hat{s}, 0)$ is a feasible solution with a lower objective value. The result then follows from the lemma. □

Thus the constrained LPD problem can be solved by application of the ALPD algorithm to the weighted LPD problem (3.5.10) with any weight factor λ greater than $\bar{\lambda}$. In general the value of $\bar{\lambda}$ is not explicitly known, so the choice of λ is open at the start of the algorithm. Numerical experience with constrained LPD problems has shown that if λ is initially chosen very large, the sequence of basic solutions encountered by the ALPD algorithm first is driven into the feasible solution set of (3.5.9) with all subsequent basic solutions remaining in this set. For sufficiently large values of λ, this behavior is guaranteed. It follows from the argument used to prove Proposition (3.5.13) that the objective values corresponding to basic solutions that are infeasible for (3.5.9) are bounded below by $\delta \cdot \lambda$, where $\delta > 0$, while objective values for feasible basic solutions do not depend on λ. Thus for sufficiently large values of λ, the latter will be uniformly lower than the former. In practice, once a feasible basic solution is attained, the value of λ can be raised at any time during the course of the algorithm to avoid a pivot operation that would result in an infeasible basic solution.

The following example demonstrates the applicability of the ALPD algorithm to the general linear programming problem.

Example (3.5.14).

The general linear programming problem is

$$\begin{aligned} & \min \ c \cdot x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \quad x \in \mathbb{R}^n \end{aligned} \tag{3.5.15}$$

where A is a $(m \times n)$ matrix assumed to be of full row rank m with $n \geq m$. The dual of (3.5.15) is

$$\begin{aligned} & \max\ b \cdot y \\ \text{s.t.} \quad & A'y \leq c \\ & y \in \mathbb{R}^m \end{aligned} \tag{3.5.16}$$

If an optimal solution $\hat{x}$ to (3.5.15) exists, then by duality theory an optimal solution $\hat{y}$ to (3.5.16) exists and $c \cdot \hat{x} = b \cdot \hat{y}$. In this case let α be any number such that $\alpha > b \cdot \hat{y}$. Then the constrained LPD problem

$$\begin{aligned} & \min\ \sigma \\ \text{s.t.} \quad & b\,y + \sigma \geq \alpha \\ & -A'y \quad\ \geq -c \\ & \sigma \geq 0 \\ & y \in \mathbb{R}^m,\ \delta \in \mathbb{R}^1 \end{aligned} \tag{3.5.17}$$

is feasible and has an optimal solution (y^*, σ^*) by Proposition (3.5.11). It is easily seen that $\sigma^* = \alpha - b \cdot \hat{y}$ and y^* is an optimal solution to the dual problem (3.5.16). A value of α need not be known explicitly for application of the ALPD algorithm. It is sufficient for purposes of calculating the required derivatives simply to consider the inequality $b \cdot y \geq \alpha$ as always being violated. The optimal solution to the primal problem (3.5.15) conveniently appears in the final tableau in the row corresponding to this inequality. □

Numerical experience reported in the next chapter suggests that if the underlying inequality system is feasible or nearly feasible, the

ALPD algorithm is competitive with standard simplex software in terms of the total number of basis changes but does not offer any significant computational advantages. For example, once a feasible basic solution to the inequality system $-A'y \geq -c$ is reached in Example (3.5.14), close examination of the ALPD algorithm reveals that for large values of the weight factor λ the pivot sequence is precisely the same as that of the dual simplex method (Dantzig [11]) applied to the primal. Thus the algorithm simply becomes a convenient method of initializing the dual simplex method if a basis with non-negative reduced costs is not readily available. However, the ALPD algorithm has shown a large computational advantage if there are a large number of infeasibilities in the inequality system. This case arises, for example, in the ℓ_1 linear approximation problem (3.5.6) since both the systems $Ax \geq b$ and $-Ax \geq -b$ appear in the LPD formulation. Similarly, the algorithm should perform well on the following constrained version of this problem.

Example (3.5.18).

Let $b_k \in \mathbb{R}^m$ be the state vector at time k for a discrete time control system governed by the equation

$$b_{k+1} = F b_k + g\alpha_k$$

where F is a $m \times m$ matrix, α_k is the scalar control applied at time k, and $g \in \mathbb{R}^m$ is a constant vector representing the change in the state vector per unit of applied control. Given b_0, the terminal error problem [19] requires the determination of a sequence of controls

$x = (\alpha_0, \alpha_1, \ldots, \alpha^{n-1})$ that minimizes the ℓ_1 norm of the difference between the terminal state vector b_n and a desired state vector b. The control sequence vector $x \in \mathbb{R}^n$ is subject to the inequality constraints

$$A_2 x \leq c$$

where A_2 is a $p \times n$ matrix and $c \in \mathbb{R}^p$.

The terminal state b_n is given by

$$b_n = F^n b_0 + F^{n-1} g\alpha_0 + F^{n-2} g\alpha_1 + \cdots + Fg\alpha_{n-2} + g\alpha_{n-1}$$

where b_0 is the initial state vector. Define the $m \times n$ matrix A_1 as

$$A_1 = (F^{n-1}g, F^{n-2}g, \ldots, Fg, g)$$

Then the terminal error problem can be formulated as the constrained least total deviations linear program

$$\begin{aligned} & \min \; e \cdot s^+ + e \cdot s^- \\ \text{s.t.} \quad & A_1 x + s^+ - s^- = b - F^n b_0 \\ & -A_2 x \qquad\qquad \geq -b_2 \\ & s^+ \geq 0, \quad s^- \geq 0 \\ & x \in \mathbb{R}^n, \quad s^+ \in \mathbb{R}^m, \quad s^- \in \mathbb{R}^m . \end{aligned}$$

□

CHAPTER 4

THE LINEARLY INSEPARABLE CASE

4.1. The Stochastic Classification Problem

In many practical applications, the patterns in a given class can be regarded as random vectors distributed according to some multivariate probability distribution. For example, in the template-matching problem defined in Section (1.2), each observed pattern in a given class is equal to the sum of one of a finite number of prototype patterns from that class and a random displacement vector attributable to random observation error or random variability in the pattern population itself. It was shown in Chapter 2 that if the underlying prototype sets are linearly separable and there exists a sufficiently small bound on the size of the random displacement vectors as measured by the ℓ_q norm, then any sets of observed patterns from the two classes are also linearly separable. If the prototype sets are completely known, then the maximum quality programs (2.3.2) and (2.3.4) with ℓ_p norm determine the linear discriminant that maximizes the bound on the ℓ_q norm of the displacement vectors while maintaining linear separability.

If, however, the prototype sets are linearly inseparable or the bounds on the displacement vectors are too large, not all sets of observed patterns will be linearly separable. More generally, let $f(x|C_1)$, $f(x|C_2)$ be probability densities corresponding to the distributions of observed patterns in class C_1 and class C_2, respectively. If these densities overlap on a region $\mathcal{R}$, where

$$\mathcal{R} = \{x \in \mathbb{R}^n : f(x|\mathcal{C}_1) > 0,\ f(x|\mathcal{C}_2) > 0\}$$

then there is no discriminant, linear or otherwise, that will always correctly classify an unknown test pattern $x \in \mathcal{R}$.

The following Bayesian model is often employed for the stochastic problem. Unknown test patterns are randomly presented from $\mathcal{C}_1$ and $\mathcal{C}_2$ with given prior probabilities of occurrence π_1 and π_2, respectively. Thus the test patterns have the mixture density

$$f(x) = \pi_1 f(x|\mathcal{C}_1) + \pi_2 f(x|\mathcal{C}_2) \tag{4.1.1}$$

Let $\Pr(\mathcal{C}_i|x)$ be the posterior probability that x belongs to $\mathcal{C}_i$, $i = 1,2$. Then by the Bayes formula,

$$\Pr(\mathcal{C}_i|x) = \pi_i f(x|\mathcal{C}_i)/f(x)\ , \qquad i = 1,2. \tag{4.1.2}$$

Define the <u>loss matrix</u>

$$L = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix}$$

where λ_{ij} is the loss incurred by deciding that an unknown test pattern belongs to $\mathcal{C}_i$ when its true class is $\mathcal{C}_j$. The expected loss for the decision "x belongs to $\mathcal{C}_i$" is thus

$$q_i(x) = \lambda_{ii} \Pr(\mathcal{C}_i|x) + \lambda_{ij} \Pr(\mathcal{C}_j|x),\quad i \neq j,\ i = 1,2 \tag{4.1.3}$$

The decision rule that minimizes the expected loss is

(4.1.4) decide $x \in \mathcal{C}_1$ if $q_1(x) \leq q_2(x)$

$x \in \mathcal{C}_2$ if $q_1(x) > q_2(x)$.

This is called the Bayes decision rule. The equivalent Bayes discriminant is

(4.1.5) $$q(x) = q_2(x) - q_1(x) .$$

Although the Bayes discriminant is optimal in the sense of minimizing the expected loss, rarely is enough information available to calculate it. The probability densities $f(x|\mathcal{C}_1)$, $f(x|\mathcal{C}_2)$ and the prior probabilities π_1 and π_2 are usually unknown. The only data available may be two given sets $\mathcal{S}_1$, $\mathcal{S}_2$ of known representatives of $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. There are several approaches in this case. First, a parametric form for each class density, such as multivariate normal, may be assumed. The sample sets $\mathcal{S}_1$, $\mathcal{S}_2$ are used to estimate the parameters and hence the density functions. The estimated density functions are combined with estimates of the prior probabilities π_1, π_2 to yield an estimate of the Bayes discriminant.

The formulational difficulty with this approach is that the assumption that a class density belongs to a known parametric family may be unwarranted. For example, in the template matching problem, the class densities may be complex mixtures of simpler densities centered around the prototypes. An alternative approach in this case is the use of non-parametric density estimation techniques such as Parzen window function estimators (Duda and Hart, [5]). The drawback

with this technique is that it can produce very complicated density function estimates that require storage of all given samples $x \in \mathcal{S}_1 \cup \mathcal{S}_2$ for implementation.

The approach taken here is to assume a parametric functional form, namely linear, for the discriminant function. The linear coefficients are chosen so that the discriminant performs well, according to some mathematical programming criterion, on given known sets of sample patterns. If the sets of sample patterns are large and well representative of their respective class populations, then the discriminant is expected to perform well on these entire class populations. The performance criterion used will be the error rate on the given sets of sample patterns. This corresponds to the Bayesian loss matrix

$$L = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The optimal Bayes discriminant is thus

$$g(x) = P(C_1|x) - P(C_2|x) \tag{4.1.6}$$

corresponding to the decision rule of assigning the pattern to the class of greater posterior probability.

4.2. Linear Discriminants by Mathematical Programming

Assume two sample sets $\mathcal{S}_1$, $\mathcal{S}_2$ of known representatives of classes C_1 and C_2, respectively, are given. Let

$$A = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix}$$

be the corresponding $m \times (n + 1)$ signed augmented pattern matrix. Then linear discriminants of the form $g(x) = w \cdot x - \theta$ can be generated as solutions to mathematical programs of the form

$$(4.2.1) \qquad \min_{u=(w,\theta) \in \mathbb{R}^{n+1}} f(u) = \sum_{i=1}^{m} f(u, a_i)$$

where $f(u,a)$ is a penalty function that reflects the performance of the discriminant defined by u on the pattern corresponding to a. Ideally, $f(u,a)$ should have the following properties.

P1. Errors should be penalized ($f(u,a) > 0$ if $a \cdot u < 0$) and correct classifications rewarded ($f(u,a) \leq 0$ if $a \cdot u > 0$).

P2. The mathematical program (4.2.1) should be easily solvable by existing algorithms.

P3. If $\mathcal{A}_1$ and $\mathcal{A}_2$ are linearly separable, the solution to (4.2.1) should determine a separating hyperplane.

These properties generally govern the choice of the function $f(a,u)$ in the models discussed below. However, in all these cases at least one of the properties has been sacrificed to achieve the others.

4.3. Minimum Error Rate Programs

If error rate is the dominant criterion for choosing a decision rule, then the best linear discriminant that can be generated from the

sample sets $\mathcal{S}_1$ and $\mathcal{S}_2$ is the one that makes the fewest misclassifications on these sets. This corresponds to the penalty function

$$f(u,a) = \begin{cases} 1 & a \cdot u \leq 0 \\ 0 & a \cdot u > 0 \end{cases}$$

Thus the objective $f(u) = \sum_{i=1}^{m} f(u,a_i)$ in (4.2.1) is equal to the number of errors made by u on $\mathcal{S}_1 \cup \mathcal{S}_2$.

Ibaraki and Maroga [8] have formulated this case as the mixed integer linear program

$$\begin{aligned} & \min\ e \cdot s \\ \text{(4.3.1)} \qquad & \text{s.t.} \quad Au + \beta I s \geq e \\ & u = (w,\theta) \in \mathbb{R}^{n+1} \\ & (s)_i = 0 \text{ or } 1, \quad i = 1,\ldots,m \end{aligned}$$

where β is a large positive number. If β is sufficiently large, then an optimal solution $(\hat{u},\hat{s})$ to (4.3.1) satisfies

$$(\hat{s})_i = 0 \quad \text{iff} \quad a_i \cdot u \geq 1$$
$$(\hat{s})_i = 1 \quad \text{iff} \quad a_i \leq 0$$

and $\hat{u}$ is thus a minimum error rate discriminant. Unfortunately, the computational difficulty of solving (4.3.1) would become prohibitive for large values of m. Thus this penalty function has properties P1 and P3 but lacks P2. Other choice of penalty function may yield a discriminant with nearly as low an error rate on $\mathcal{S}_1 \cup \mathcal{S}_2$ with far less computational effort.

4.4. Least Squares Programs

The penalty function choice $f(u,a) = (1-a\cdot u)^2$ results in the program

$$\min_{u=(w,\theta)\in\mathbb{R}^{n+1}} \|Au - e\|_2^2 \tag{4.4.1}$$

which is an example of the linear approximation problem with ℓ_2 norm discussed in Section (3.5). As noted there, the explicit solution to (4.4.1) is

$$u = (A'A)^{-1}A'e \tag{4.4.2}$$

where the existence of $(A'A)^{-1}$ is guaranteed by the assumption that A is of full column rank $(n + 1)$. Computationally, this is the easiest of the models to solve. However, the model lacks Properties P1 and P3. The function $(1 - a\cdot u)^2$ penalizes both incorrect $(a\cdot u \leq 0)$ and correct $(a\cdot u > 0)$ classifications. For correct classifications, the penalty actually increases as $a\cdot u$ increases past the margin value of one. The following simple example illustrates the absence of Property P3 due to this unfortunate behavior.

Example (4.4.3).

Let $\mathcal{A}_1 = \{\alpha,2,1\}$, $\mathcal{A}_2 = \{-1\}$ be one-dimensional pattern sets with $\alpha > 0$. Clearly $\mathcal{A}_1$ and $\mathcal{A}_2$ are linearly separable by the discriminant $g(x) = x$ for all positive values of α. The signed augmented pattern matrix is

$$A = \begin{bmatrix} \alpha & -1 \\ 2 & -1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$$

A direct calculation using (4.4.2) shows that the least squares discriminant, after normalization to make the coefficient of x equal to unity, is given by

$$g(x) = x + \frac{\alpha^2 - 6\alpha + 4}{2\alpha + 12}$$

For all values $\alpha \geq 4 + 2\sqrt{6}$, $g(-1) \geq 0$ and hence the pattern in $\mathcal{S}_2$ is misclassified. The penalty that the least squares criterion places on excessively large absolute values of the discriminant function for both correct and incorrect classifications gives too much influence to isolated patterns that are far from the main group. □

Despite this drawback, the least squares discriminant has a significant asymptotic property. Patterson and Womack [20] show that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are constructed from class $\mathcal{C}_1$ and $\mathcal{C}_2$ patterns, respectively, by selecting m independent patterns of known classification from the mixture distribution with density

$$f(x) = \pi_1 f(x|\mathcal{C}_1) + \pi_2 f(x|\mathcal{C}_2) ,$$

then the discriminant defined by (4.4.2) asymptotically approaches the minimum squared error approximation to the Bayes discriminant $g_0(x)$

as $m \to \infty$. This approximation minimizes

$$\int [(w \cdot x - \theta) - g_0(x)]^2 \, f(x)dx \tag{4.4.4}$$

However, as Duda and Hart [5] point out, the best linear approximation to the Bayes discriminant does not necessarily have any favorable error rate properties. Points where $f(x)$ is large and points far from the surface $g_0(x) = 0$ are emphasized at the expense of points near this surface.

4.5. Linear Discriminants by Least Positive Deviations

The penalty function

$$f(u,a) = (1 - a \cdot u)^+ \tag{4.5.1}$$

leads to the LPD linear program first suggested in a pattern classifi- context by Smith [21]

$$\begin{aligned} &\min \; e \cdot s \\ \text{s.t.} \quad &Au + Is \geq e \\ &s \geq 0 \\ &u = (w, \theta) \in \mathbb{R}^{n+1}, \quad s \in \mathbb{R}^m \end{aligned} \tag{4.5.2}$$

This model has the property P2 since it is relatively easy to solve by the ALPD algorithm presented in Chapter 3 or by the simplex method with upper bounds applied to the dual. In addition, property P3 is satisfied since linearly separable problems are characterized by the feasibility

of the system $Au \geq e$, so (4.5.2) will produce a separating hyperplane if one exists. The only property not satisfied is P1. A penalty is incurred whenever $f(a,u) > 0$, which is equivalent to the event $a \cdot u < 1$. This event will be called a <u>margin violation</u>. A margin violation is a true misclassification only if $a \cdot u \leq 0$. Thus correct classifications are penalized if $0 < a \cdot u < 1$.

The dual of (4.5.2) is

$$\begin{aligned} &\max\ e \cdot y \\ &\text{s.t.}\quad A'y = 0 \\ &\quad 0 \leq y \leq e \\ &\quad y \in \mathbb{R}^m \end{aligned} \tag{4.5.3}$$

Let $\hat{y}$ be an optimal basic solution to (4.5.3) as determined by the simplex method with upper bounds, and let $\hat{u}$ be the optimal primal solution which is the simplex multiplier vector for the terminal optimal basis in (4.5.3). Assuming non-degeneracy, the termination conditions (3.1.8) for each non-basic variable $(\hat{y})_i$ are

$$\begin{aligned} (\hat{y})_i = 0 &\iff a_i \cdot u > 1 \\ (\hat{y})_i = 1 &\iff a_i \cdot u < 1 \end{aligned} \tag{4.5.4}$$

Thus the patterns which are margin violators are those for which the corresponding optimal dual variables are at the upper bound.

Two distinct cases arise which are distinguished by the form of the optimal solution $\hat{u} = (\hat{w}, \hat{\theta})$:

Case 1. $\hat{w} = 0$. This case occurs when the corresponding optimal dual basis in (4.5.3) consists of signed augmented patterns which are all derived from a single sample set $\mathcal{S}_1$ or $\mathcal{S}_2$. The sample set which is the source of the optimal dual basis will be called dominant. If $\mathcal{S}_1$ is dominant, $\hat{\theta} = -1$; otherwise $\hat{\theta} = +1$. In general the dominant set is the larger of the two sample sets since the optimal objective value is equal to twice the number of patterns in the non-dominant set. The discriminant corresponding to $\hat{u}$ is the constant function $f(x) = -\hat{\theta}$, which is equivalent to the decision rule that classifies all patterns into the class of the dominant sample set. Since all of the inequalities corresponding to patterns in the dominant set are tight at $u = \hat{u}$, this solution is degenerate if there are more than $(n + 1)$ such patterns.

This case may arise, for example, when one of the sample sets is overwhelmingly larger than the other and the sets are not linearly separable. In such circumstances the discriminant $f(x) = -\hat{\theta}$, although uninteresting, is not unreasonable. However, this case can be avoided if desired by appending additional constraints to (4.5.2) as seen in several models discussed below or by solving a weighted problem where greater weight is assigned to the smaller set $\mathcal{S}_i$ (see 4.5.8).

Case 2. $\hat{w} \neq 0$. This is the case of interest which occurs when the optimal dual basis consists of a mixture of signed augmented patterns derived from both pattern sets. The discriminant hyperplane can

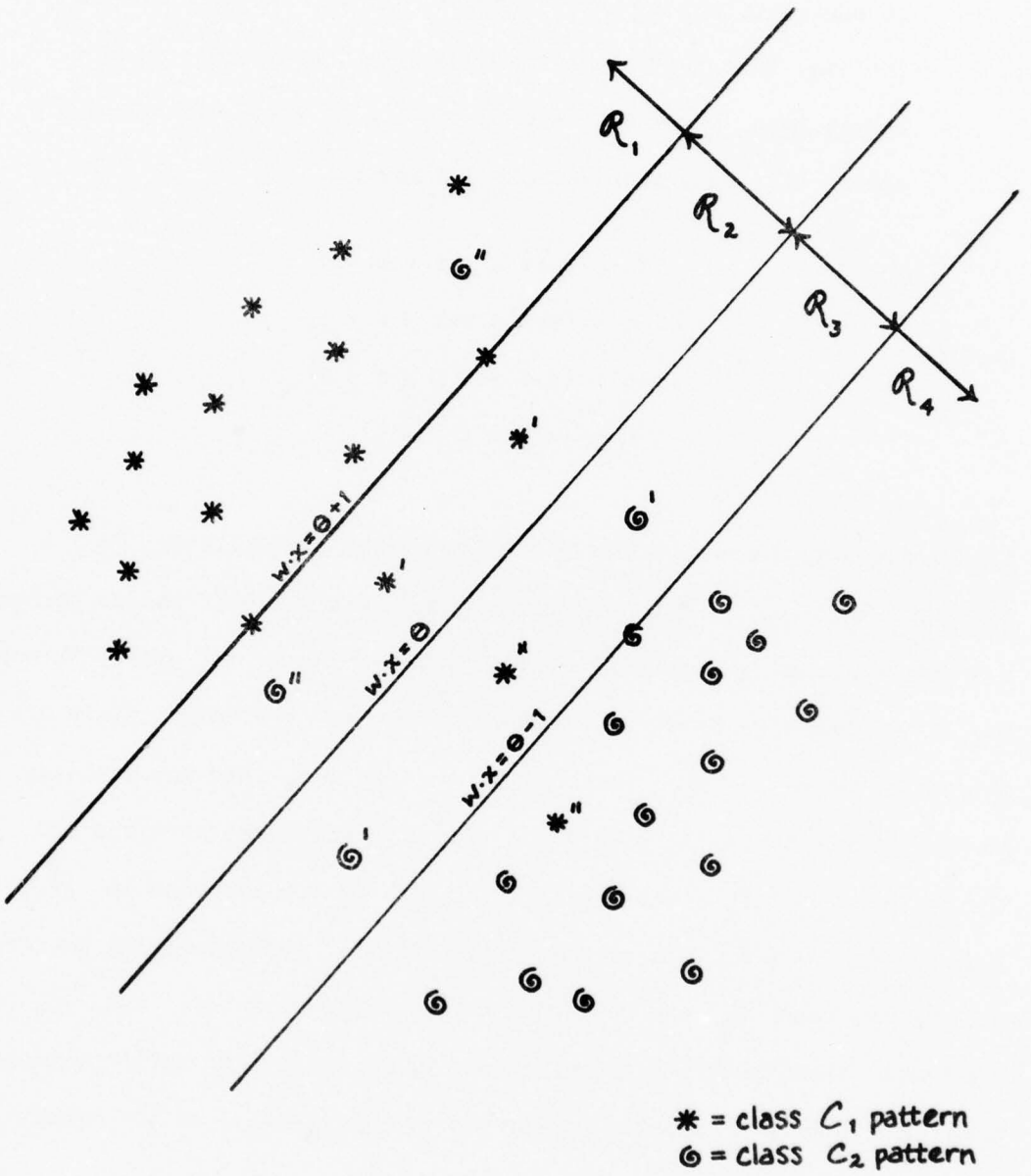


Figure (4.5.5). LPD Discriminant and Margin Planes Divide the Pattern Space into Four Regions. Correctly Classified Margin Violators Are Marked with a (') and True Misclassifications with a (").

be characterized geometrically as follows. As shown in Figure (4.5.5), this hyperplane together with the parallel margin hyperplanes $\hat{w}\cdot x = \hat{\theta}-1$ and $\hat{w}\cdot x = \hat{\theta} + 1$ divide the pattern space $\mathbb{R}^n$ into four regions defined by

$$
\begin{aligned}
\mathcal{R}_1 &= \{x:\hat{w}\cdot x \geq \hat{\theta} + 1\} \\
\mathcal{R}_2 &= \{x:\hat{\theta} \leq w\cdot x < \hat{\theta} + 1\} \\
\mathcal{R}_3 &= \{x:\hat{\theta} - 1 < w\cdot x \leq \hat{\theta}\} \\
\mathcal{R}_4 &= \{x:\hat{w}\cdot x \leq \hat{\theta} - 1\}
\end{aligned}
\tag{4.5.6}
$$

Assuming non-degeneracy, exactly $(n + 1)$ of the inequalities $Au \geq e$ are tight at $u = \hat{u}$. Thus the margin plane $w\cdot x = \theta + 1$ passes through k patterns from $\mathcal{S}_1$ and the margin plane $w\cdot x = \theta - 1$ passes through $n + 1 - k$ patterns from $\mathcal{S}_2$ where $1 \leq k \leq n$. The margin violators are the patterns in $\mathcal{S}_1$ that lie in $\mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$ and the patterns in $\mathcal{S}_2$ that lie in $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$. The true misclassifications are the patterns in $\mathcal{S}_1$ that lie in $\mathcal{R}_3 \cup \mathcal{R}_4$ and the patterns in $\mathcal{S}_2$ that lie in $\mathcal{R}_1 \cup \mathcal{R}_2$. It is shown below that if the underlying pattern classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are bounded and the sample sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are large, then there are approximately equal numbers of margin violators from $\mathcal{S}_1$ and $\mathcal{S}_2$ and the centers of gravity (means) of the margin violators in each set are approximately equal.

The following example illustrates these concepts

Example (4.5.7).

Let $\mathcal{S}_1 = \{1,2,\ldots, k, -(k+1)\}$, $\mathcal{S}_2 = \{-1,-2,\ldots,-k, (k+1)\}$ be one-dimensional pattern sets with $k \geq 2$. This is the same problem

as that discussed in Example (2.4.6) where the poor performance of the maximum quality hyperplane was revealed. Here the LPD discriminant will be shown to be $g(x) = \alpha x$ where $\alpha > 0$. This discriminant correctly classifies all patterns in $\mathcal{S}_1 \cup \mathcal{S}_2$ except the outliers $-(k+1)$ in $\mathcal{S}_1$ and $(k+1)$ in $\mathcal{S}_2$.

The signed augmented pattern matrix is

$$A = \begin{bmatrix} 1 & -1 \\ 2 & -1 \\ \vdots & -1 \\ k & -1 \\ -(k+1) & -1 \\ 1 & 1 \\ 2 & 1 \\ \vdots & 1 \\ k & 1 \\ -(k+1) & 1 \end{bmatrix}$$

For $i = 1,2,\ldots,k$, the two signed augmented patterns $(i,-1)$ and $(i,1)$ derived from $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, define the basic inequality solution $u_i = (1/i,0)$. As seen in Section (3.3), u_i defines an optimal solution to (4.5.2) if

$$\left.\begin{aligned} \frac{df}{d\tau^+}(u_i + \tau d_j)\Big|_{\tau=0} &\geq 0 \\ \frac{df}{d\tau^-}(u_i + \tau d_j)\Big|_{\tau=0} &\leq 0 \end{aligned}\right\} \; j = 1,2$$

where the directions

$$d_1 = \begin{pmatrix} 1/(2i) \\ -1/2 \end{pmatrix} \qquad \text{and} \qquad d_2 = \begin{pmatrix} 1/(2i) \\ 1/2 \end{pmatrix}$$

are the first and second columns of the matrix

$$\begin{pmatrix} i & -1 \\ i & 1 \end{pmatrix}^{-1}$$

and f is the LPD objective function defined by $f(u) = e \cdot (e - Au)^+$. A direct calculation from formulas (3.2.8) yields

$$\frac{df}{d\tau^+}(u_i + \tau d_1)\Big|_{\tau=0} = \frac{df}{d\tau^+}(u_i + \tau d_2)\Big|_{\tau=0}$$

$$= \frac{1}{i}\left[\frac{i(i+1)}{2} - (k+1)\right]$$

and

$$\frac{df}{d\tau^-}(u_i + \tau d_1)\Big|_{\tau=0} = \frac{df}{d\tau^-}(u_i + \tau d_2)\Big|_{\tau=0}$$

$$= \frac{1}{i}\left[\frac{(i-1)(i)}{2} - (k+1)\right]$$

Let i^* be the smallest positive integer such that

$$\frac{i^*(i^* + 1)}{2} \geq k + 1$$

For $k \geq 2$, it is easily verified that $1 < i^* \leq k$ and hence u_{i^*} is optimal for (4.5.2). The corresponding discriminant is $g(x) = (1/i^*)x$. The margin violators for this discriminant consist of all patterns x such that $|x| < i^*$ and the two outliers, while the only true misclassifications are those two outliers. □

If misclassifications of patterns from one class are considered more serious than misclassifications from the other class, it may be desirable to adjust the penalty functions accordingly. The LPD program (4.5.2) can be generalized to the weighted LPD model

$$
\begin{aligned}
&\min \quad \alpha_1 e \cdot s_1 + \alpha_2 e \cdot s_2 \\
\text{(4.5.8)} \qquad &\text{s.t.} \quad A_1 u + I s_1 \geq e \\
&\qquad\qquad A_2 u + I s_2 \geq e \\
&\qquad\qquad s_1 \geq 0, \quad s_2 \geq 0 \\
&u = (w, \theta) \in \mathbb{R}^{n+1}, \qquad (s_1, s_2) \in \mathbb{R}^m
\end{aligned}
$$

where A_1 and A_2 are the signed augmented pattern matrices for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, and α_1 and α_2 are scalar weighting factors reflecting the relative penalty on each type of error. The dual of (4.5.8) is

$$
\begin{aligned}
&\max \quad e \cdot y_1 + e \cdot y_2 \\
\text{(4.5.9)} \qquad &\text{s.t.} \quad A_1' y_1 + A_2' y_2 = 0 \\
&\qquad\qquad 0 \leq y_1 \leq \alpha_1 e, \quad 0 \leq y_2 \leq \alpha_2 e \\
&\qquad\qquad y = (y_1, y_2) \in \mathbb{R}^m
\end{aligned}
$$

Let $\hat{y} = (\hat{y}_1, \hat{y}_2)$ be an optimal solution to (4.5.9) as determined by the simplex method with upper bounds and let $\hat{u}$ be the corresponding simplex multiplier vector that defines the weighted LPD solution to (4.5.8). Define the following index sets

$B_j = \{i:(\hat{y})_i$ is basic and $x_i \in \mathcal{S}_j\}$, $j = 1,2$

$U_j = \{i:(\hat{y})_i$ is non-basic, equal to its upper bound of α_j, and $x_i \in \mathcal{S}_j\}$, $j = 1,2$

$L = \{i:(\hat{y})_i$ is non-basic and equal to its lower bound of zero$\}$

The termination criteria for the simplex method with upper bounds imply that U_1 and U_2 are the index sets of the margin violators from $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Let m_1 and m_2 be the respective numbers of elements in U_1 and U_2, and let

$$\bar{x}_j = \frac{1}{m_j} \sum_{i \in U_j} x_i , \qquad j = 1,2,$$

be the mean of the margin-violators from $\mathcal{S}_j$. The following proposition will be used to show that for large values of m_1 and m_2, the ratio m_1/m_2 of the numbers of margin violators from $\mathcal{S}_1$ to margin violators from $\mathcal{S}_2$ is approximately equal to the inverse penalty ratio α_2/α_1. Furthermore, if both pattern classes C_1 and C_2 are bounded, then the means $\bar{x}_1$ and $\bar{x}_2$ are approximately equal.

PROPOSITION (4.5.10). Let $\hat{\lambda}$ be the optimal objective value corresponding to $\hat{u}$ in (4.5.8) and let $\gamma = \max_{i=1,\ldots,m} \|x_i\|$ for any vector norm $\|\cdot\|$. Then

a) $\hat{\lambda} - (n + 1) \max(\alpha_1,\alpha_2) \leq m_1\alpha_1 + m_2\alpha_2 \leq \hat{\lambda}$

b) $|m_1\alpha_1 - m_2\alpha_2| \leq (n + 1) \max(\alpha_1,\alpha_2)$

c) $\|\bar{x}_1 - \bar{x}_2\| \leq 2(n + 1)\, \gamma[\max(\alpha_1,\alpha_2)/\max(\alpha_1 m_1, \alpha_2 m_2)]$

Proof. Equality of the optimal primal and dual objective values implies

$$\hat{\lambda} = \sum_{i \in B_1 \cup B_2} (\hat{y})_i + \sum_{i \in U_1 \cup U_2} (\hat{y})_i + \sum_{i \in L} (\hat{y})_i$$

By the termination criteria of the simplex method with upper bounds, the second term is equal to $\alpha_1 m_1 + \alpha_2 m_2$ and the third term vanishes. Part a) then follows immediately from the bounds on the (n+1) basic variables in the optimal dual solution. Substitution of the known values of the non-basic optimal dual variables in the constraint set $A_1' y_1 + A_2' y_2 = 0$ yields

$$(4.5.11) \quad \alpha_1 \sum_{i \in U_1} \begin{pmatrix} x_i \\ -1 \end{pmatrix} - \alpha_2 \sum_{i \in U_2} \begin{pmatrix} x_i \\ -1 \end{pmatrix} = \sum_{i \in B_2} \begin{pmatrix} x_i \\ -1 \end{pmatrix} (\hat{y})_i - \sum_{i \in B_1} \begin{pmatrix} x_i \\ -1 \end{pmatrix} (\hat{y})_i$$

The last of the (n+1) equations (4.5.11) implies

$$|m_2 \alpha_2 - m_1 \alpha_1| = \left| \sum_{i \in B_1} (\hat{y})_i - \sum_{i \in B_2} (\hat{y})_i \right|$$

$$\leq \sum_{i \in B_1 \cup B_2} (\hat{y})_i$$

Part b) then follows from the bounds on the (n+1) elements in $B_1 \cup B_2$. Application of the triangle inequality to the first n equations in (4.5.11) yields

$$(4.5.12) \qquad \|\alpha_1 m_1 \bar{x}_1 - \alpha_2 m_2 \bar{x}_2\| \leq (n+1)\gamma \max(\alpha_1, \alpha_2)$$

But

$$\|\bar{x}_1 - \bar{x}_2\| = \left\| \bar{x}_1 - \frac{\alpha_2 m_2}{\alpha_1 m_1} \bar{x}_2 + \frac{\alpha_2 m_2}{\alpha_1 m_1} \bar{x}_2 - \bar{x}_2 \right\|$$

$$\leq \left\| \bar{x}_1 - \frac{\alpha_2 m_2}{\alpha_1 m_1} \bar{x}_2 \right\| + \frac{|\alpha_2 m_2 - \alpha_1 m_1|}{\alpha_1 m_1} \|\bar{x}_2\|$$

By (4.5.12),

$$\left\| \bar{x}_1 - \frac{\alpha_2 m_2}{\alpha_1 m_1} \bar{x}_2 \right\| \leq \frac{(n+1)\gamma \max(\alpha_1, \alpha_2)}{\alpha_1 m_1}$$

and by part b)

$$\frac{|\alpha_2 m_2 - \alpha_1 m_1|}{\alpha_1 m_1} \|\bar{x}_2\| \leq \frac{(n+1)\gamma \max(\alpha_1, \alpha_2)}{\alpha_1 m_1}$$

Thus

$$\|\bar{x}_1 - \bar{x}_2\| \leq \frac{2(n+1)\gamma \max(\alpha_1, \alpha_2)}{\alpha_1 m_1} \tag{4.5.13}$$

Part c) then follows from (4.5.13) and the symmetrical relation obtained by reversing the roles of $\bar{x}_1$ and $\bar{x}_2$. □

COROLLARY (4.5.14). If the underlying pattern classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are bounded, then

$$\lim_{m_1, m_2 \to \infty} \begin{pmatrix} \dfrac{\hat{\lambda}}{m_1\alpha_1 + m_2\alpha_2} \\ \dfrac{m_1}{m_2} \\ \|\bar{x}_1 - \bar{x}_2\| \end{pmatrix} = \begin{pmatrix} 1 \\ \dfrac{\alpha_2}{\alpha_1} \\ 0 \end{pmatrix}$$

□

Several additional models have been proposed that eliminate the use of a margin vector and hence the distinction between margin violators and true misclassifications. In general these are constrained weighted LPD problems of the form

$$\begin{aligned} &\min\ u\cdot s_1 + v\cdot s_2 \\ \text{(4.5.15)}\qquad &\text{s. t.}\quad A_1u + Is_1 \geq 0 \\ &\qquad\quad A_2u + Is_2 \geq 0 \\ &\qquad\quad Gu \geq b \\ &\qquad\quad s_1 \geq 0,\quad s_2 \geq 0 \\ &\qquad\quad u = (w,\theta) \in \mathbb{R}^{n+1},\quad (s_1,s_2) \in \mathbb{R}^m \end{aligned}$$

where u and v are strictly positive weight vectors and $Gu \geq b$ is a set of added constraints that eliminate the useless solution $u = 0$ and the uninteresting solution $w = 0$, $\theta = \pm 1$ that occurs when one of the pattern sets is dominant. Grinold [6] suggests a single added constraint of the form

$$g^*\cdot u \geq 1$$

where $g^* = e\cdot A/m$, i.e. the mean of all the signed augmented patterns. The program (4.5.15) will be feasible as long as $g^* \neq 0$. (The case $g^* = 0$ occurs only when the numbers of sample patterns from each class are equal and the sample means are equal.) Another possibility is the pair of constraints

$$g_1^*\cdot u \geq 1$$

$$g_2^*\cdot u \geq 1$$

where g_j^*, $j = 1,2$ is the mean of the signed augmented sample patterns from class j. For the sake of feasibility it is required that $g_1^* \neq -g_2^*$ or equivalently that the sample means of the two classes differ. The linear discriminant produced by (4.5.15) then separates the two class sample means.

4.6. A Numerical Experiment

In order to compare the behavior of the ALPD and simplex algorithms under various problem conditions, the following numerical experiment was devised.

Two n-dimensional pattern sets $\mathcal{S}_1$ and $\mathcal{S}_2$ were constructed, each containing $m/2$ patterns. Each pattern x in $\mathcal{S}_1$ was generated by the formula

$$(x)_i = u[-1/2, 1/2] - \lambda , \qquad i = 1,\ldots,n$$

where $u[-1/2, 1/2]$ is a pseudorandom number uniformly distributed in the interval $[-1/2, 1/2]$. Thus the patterns in $\mathcal{S}_1$ are pseudorandom vectors uniformly distributed in the interior of the unit n-dimensional hypercube H_1 centered at $-(\lambda,\ldots,\lambda)$. Similarly, the patterns in $\mathcal{S}_2$ were generated by the formula

$$(x)_i = u[-1/2, 1/2] + \lambda , \qquad i = 1,\ldots,n$$

These pseudorandom vectors are uniformly distributed in the interior of the unit n-dimensional hypercube H_2 centered at $(\lambda,\ldots,\lambda)$. The situation for $n = 2$ is illustrated in Figure (4.6.1).

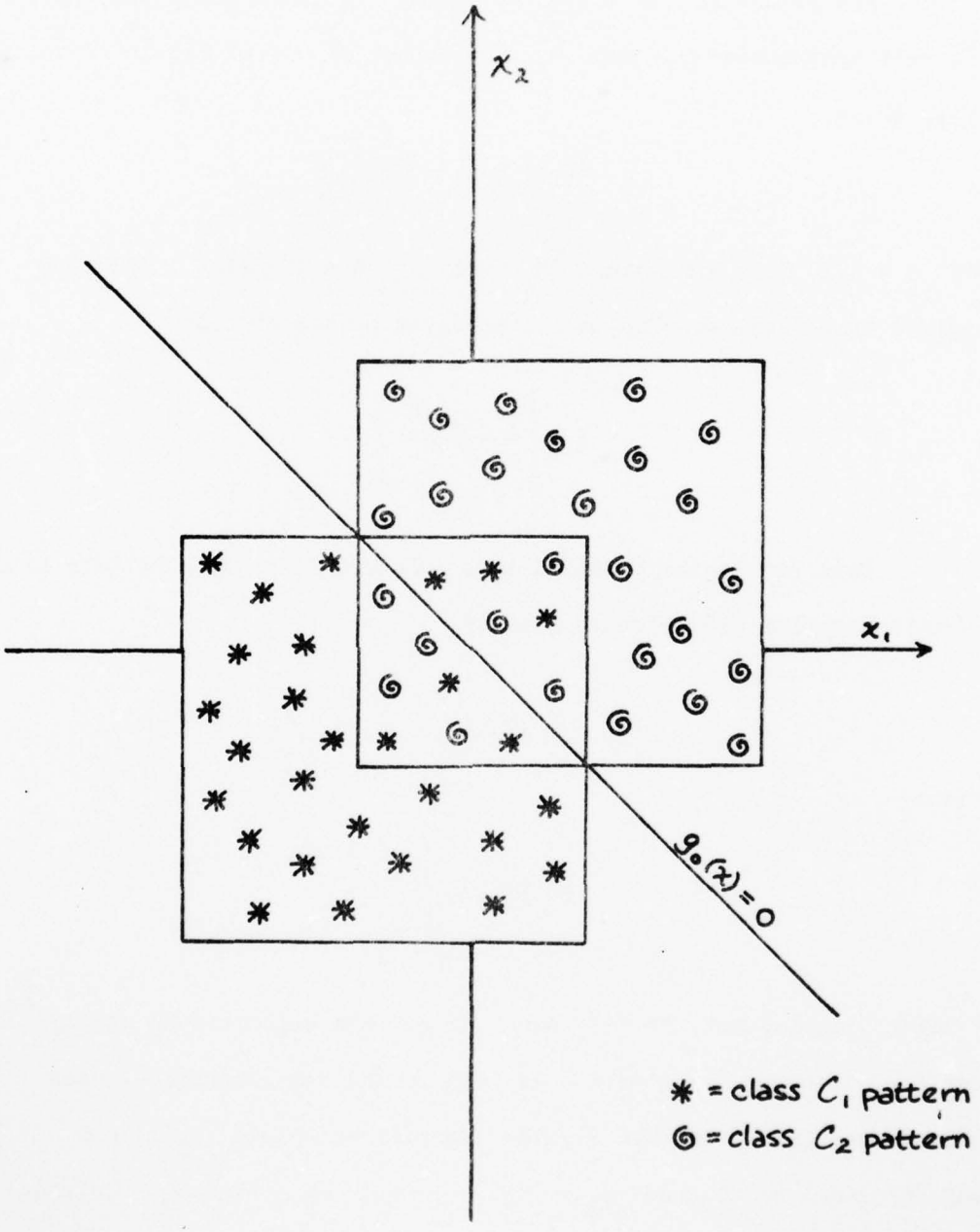


Figure (4.6.1). Overlapping Hypercube Problem for n = 2 With Bayes Discriminant $g_0(x) = - x_1 - x_2$.

For values of the scalar parameter λ in the interval $[0,1/2]$, the unit hypercubes H_1 and H_2 overlap on a cubical region of volume $V(\lambda)$, where

$$V(\lambda) = (1 - 2\lambda)^n.$$

For $\lambda > 1/2$, $V(\lambda) = 0$ since H_1 and H_2 are disjoint. Thus any desired fractional overlap α is achieved by the setting

$$\lambda = \left[\frac{1 - \alpha^{1/n}}{2} \right]$$

This problem is intended to simulate a stochastic pattern classification problem with mixture density

$$f(x) = \frac{1}{2} f(x|C_1) + \frac{1}{2} f(x|C_2)$$

where

$$f(x|C_i) = \begin{cases} 1 & \text{if } x \in H_i \\ 0 & \text{otherwise} \end{cases}$$

A Bayes discriminant for the lowest error rate criterion is easily verified to be $g_0(x) = -e \cdot x$. As seen in the two-dimensional case illustrated in Figure (4.6.1), the discriminant plane $g_0(x) = 0$ separates all of classes $C_1 = \{x \in H_1\}$ and $C_2 = \{x \in H_2\}$ outside the region of overlap and passes through the center of this region, misclassifying exactly half of each class there for an overall error rate of $\alpha/2$.

A series of pattern set pairs $\mathcal{S}_1$, $\mathcal{S}_2$ was generated for various values of the total number of patterns m, the pattern dimensionality n, and the fractional overlap α. All combinations of the parameter values

$$\begin{aligned} m &= 100, 200, 500, 1000 \\ n &= 1, 2, 5, 10 \\ \alpha &= 0, .2, .4, .6, .8, 1.0 \end{aligned} \tag{4.6.2}$$

were used for a total of 96 cases. Usually five independent test problems were run for each case, although only two and in some cases one problem were run for some of the larger values of m and n. Altogether a total of 377 independent problems were solved.

For each case, the signed augmented pattern matrix A was constructed. The LPD problem (4.5.2) with tableau [A:e] was solved with the ALPD algorithm, while the dual (4.5.3) was solved by the simplex method with upper bounds (SMUB). Since the two algorithms use identical pivot operations for basis changes but differ only in the pivot selection rules, the total number of pivots (basis changes) required to reach an optimum solution from the same initial, arbitrarily chosen basis serves as a convenient basis of comparison. (Thus the ALPD initialization algorithm given in Section (3.4) was not used. Rather the (n+1) members of the initial basis were chosen as the signed augmented patterns corresponding to the first n patterns in $\mathcal{S}_1$ and the first pattern in $\mathcal{S}_2$.)

In all cases except those for which $\alpha = 0$, the same optimal basis was achieved by both algorithms. When $\alpha = 0$ the pattern sets are linearly separable and several distinct optimal bases may exist. Frequently the algorithms arrived at different optimal solutions in this case, although of course each optimal solution defined a separating hyperplane. In general, the error rate achieved on $\mathcal{A}_1 \cup \mathcal{A}_2$ by the discriminant corresponding to the optimal solution was usually very close to the Bayes error rate of $\alpha/2$ with small fluctuations about this rate due to the finite size of the pattern sets.

Average values of the numbers of pivots required by the ALPD and SMUB algorithms are listed in Table (4.6.3) for each case. Some graphical representation of this data is provided by Figures (4.6.4) through (4.6.13) which reveal two clear trends.

First, as seen in Figures (4.6.4) through (4.6.7), the SMUB algorithm is highly sensitive to the fractional overlap α while the ALPD algorithm is not. For $\alpha = 0$ the numbers of required SMUB and ALPD pivots are nearly equal. As α and here the degree of infeasibility of the system $Au \geq e$ increases, the number of SMUB pivots increases very quickly and then levels off while the number of ALPD pivots remains relatively constant. For several cases with large values of α, the relative advantage of the ALPD algorithm in terms of number of pivots reaches a factor of several hundred. For a given value of α, this factor seems to be an increasing function of the aspect ratio $m/(n+1)$ of the matrix A.

Second, as seen in Figures (4.6.8) through (4.6.13), for fixed values of α and n the number of pivots appears to be a linearly increasing function of m. However, except for the $\alpha = 0$ case, the rate of increase is much higher for the SMUB than the ALPD algorithms.

The computational advantage of the ALPD algorithm thus appears most significant for problems in which the matrix A has a high aspect ratio and the underlying inequality system has a large degree of infeasibility. Such problems arise not only in linearly inseparable pattern classification models with large pattern sets but also in the linear approximation problem (3.5.6) with ℓ_1 norm. For such problems usually the number of data points greatly exceeds the number of parameters to be fit, thus creating the high aspect ratio situation. The large degree of infeasibility in the underlying inequality system arises naturally since it is comprised of the two systems $Ax \geq b$ and $-Ax \geq -b$.

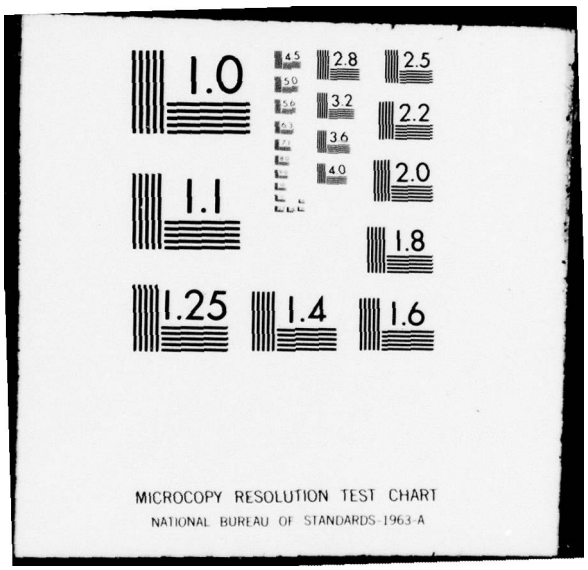
Problem Size (m,n)	Percent Overlap 0% ALPD	0% SMUB	20% ALPD	20% SMUB	40% ALPD	40% SMUB	60% ALPD	60% SMUB	80% ALPD	80% SMUB	100% ALPD	100% SMUB
(100,1)	2.2	2.6	3.2	22.4	3.0	58.0	2.8	83.2	2.4	108.6	3.6	116.8
(200,1)	5.0	3.0	2.8	56.2	3.2	119.2	2.4	167.4	2.6	222.8	4.0	221.6
(500,1)	6.6	3.0	4.0	147.8	3.2	262.2	3.2	437.8	3.2	565.0	3.4	556.4
(1000,1)	6.0	3.0	4.6	279.4	4.2	571.2	3.5	870.0	3.5	1125.0	3.5	1100.5
(100,2)	4.4	5.0	6.8	35.6	6.8	58.2	7.0	88.2	4.6	109.0	6.6	123.2
(200,2)	8.0	7.0	7.8	62.4	8.6	131.2	8.6	172.8	6.8	214.4	6.6	237.6
(500,2)	8.2	6.8	9.8	176.2	9.4	297.2	9.0	422.6	8.8	556.2	7.0	611.0
(1000,2)	10.5	6.5	9.5	324.0	14.0	615.5	11.5	826.0	10.5	1083.0	10.5	1208.0
(100,5)	10.2	12.8	16.4	68.0	18.6	94.0	14.2	115.6	13.4	125.0	16.4	135.0
(200,5)	16.0	16.8	17.2	139.4	18.0	181.0	15.6	237.0	18.4	261.0	17.4	276.8
(500,5)	21.0	35.5	26.5	319.5	29.5	494.5	19.5	592.0	23.5	680.5	22.0	690.0
(1000,5)	22.5	13.5	24.5	637.5	24.0	957.5	27.0	1156.0	20.0	1289.0	18.0	1395.0
(100,10)	26.8	27.8	24.0	97.4	26.8	121.2	29.0	131.2	30.0	148.6	25.4	150.4
(200,10)	23.4	45.0	33.0	179.2	37.8	269.0	34.2	297.8	36.2	311.8	34.2	307.2
(500,10)	34.0	50.0	45.5	485.5	37.5	633.0	36.0	730.5	45.0	777.0	40.0	795.0
(1000,10)	41.0	68.0	44.0	916.0	55.0	1317.0	52.0	1498.0	53.0	1504.0	55.0	1587.0

Table (4.6.3). Average Number of Pivots Required by Accelerated Least Positive Deviations (ALPD) and Simplex Method with Upper Bounds (SMUB) Algorithms

AD-A061 496 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 5/8
MATHEMATICAL PROGRAMMING APPLICATIONS IN PATTERN RECOGNITION.(U)
AUG 78 R H LEARY N00014-75-C-0267
UNCLASSIFIED SOL-78-14 NL

2 OF 2
AD
A061496

END
DATE
FILMED
2-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

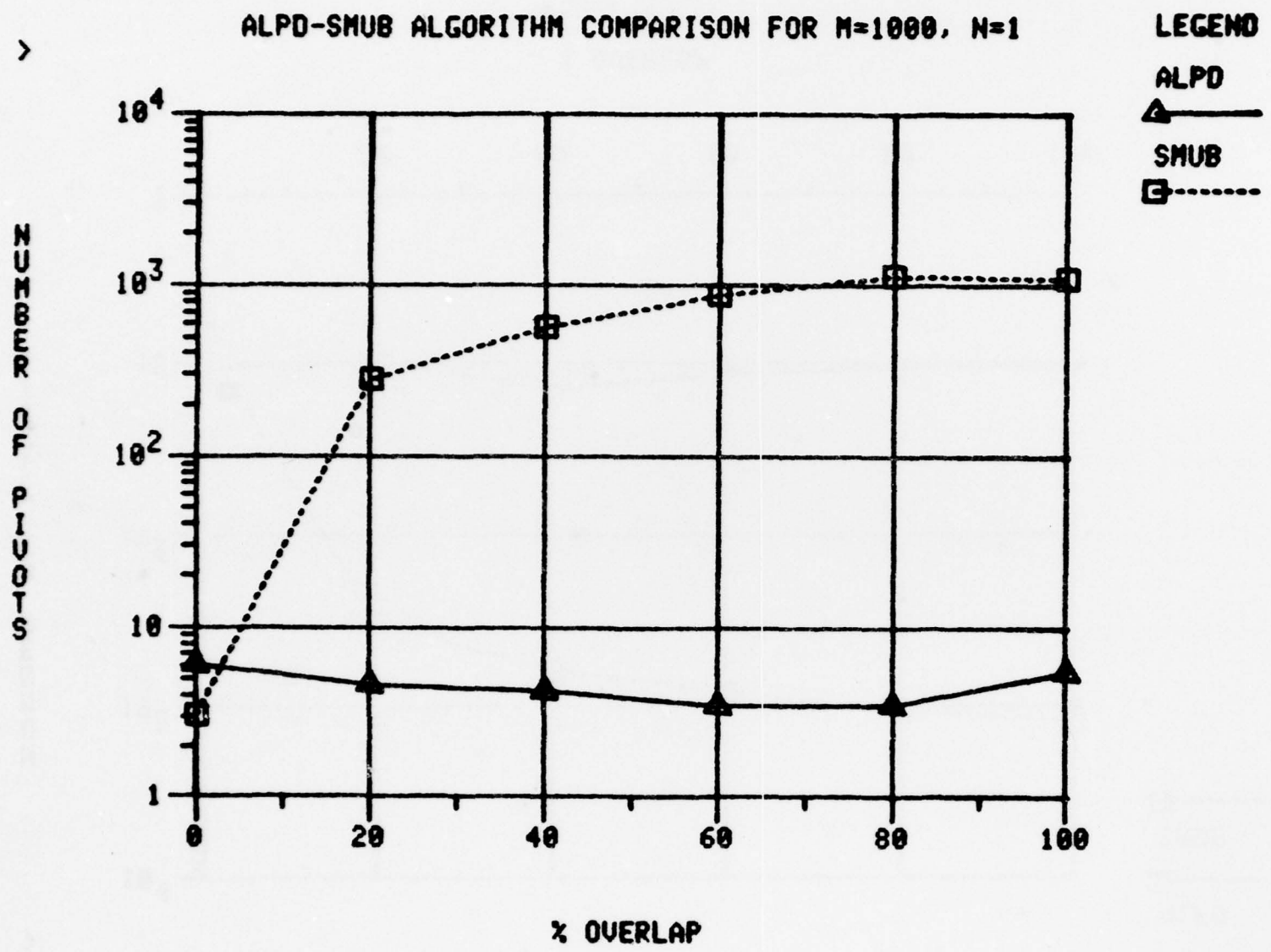

ALPD-SMUB ALGORITHM COMPARISON FOR M=1000, N=1
LEGEND
ALPD
SMUB
NUMBER OF PIVOTS
10^4
10^3
10^2
10
1
0
20
40
60
80
100
% OVERLAP

Figure (4.6.4)

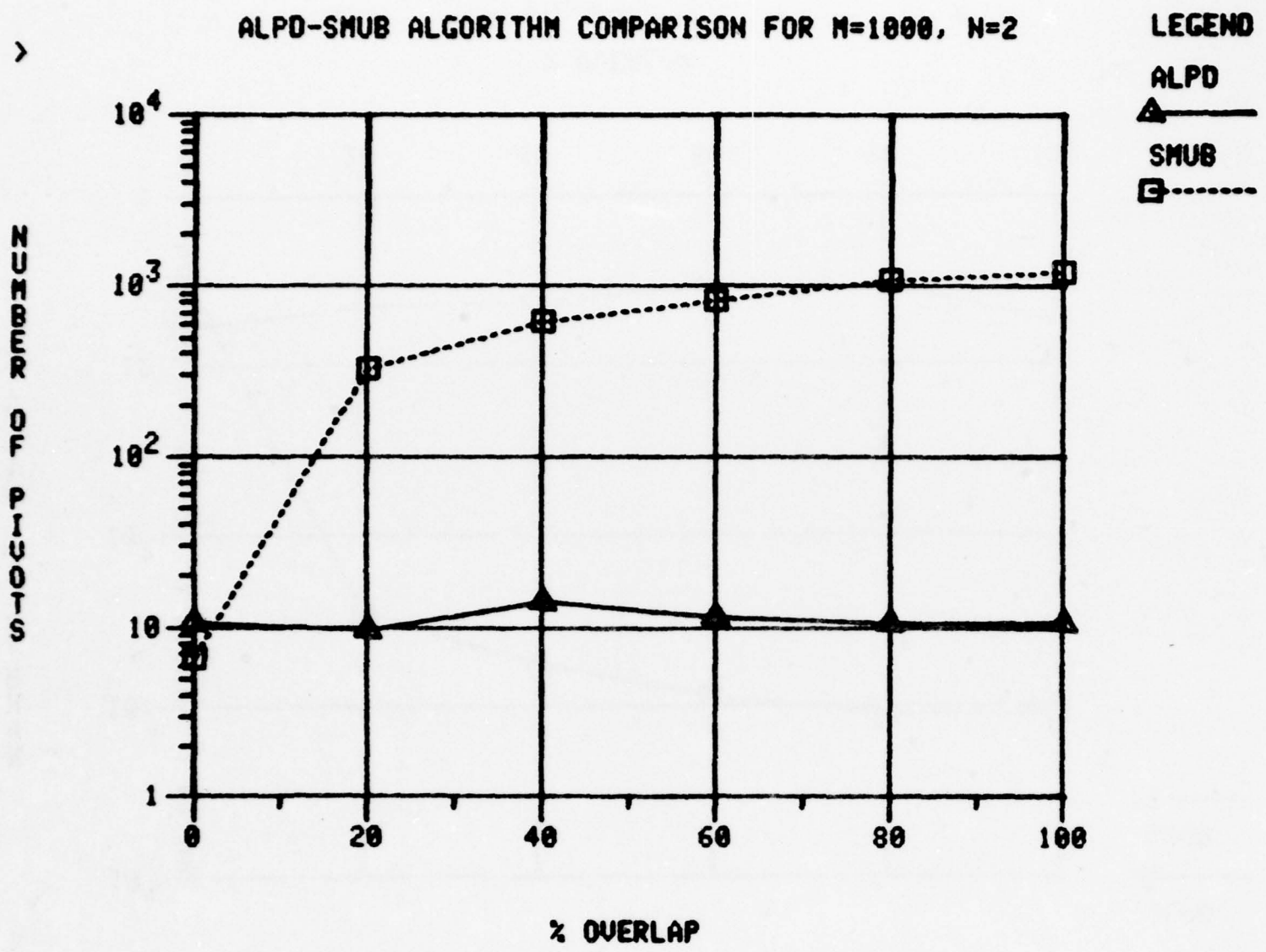

ALPD-SMUB ALGORITHM COMPARISON FOR M=1000, N=2
LEGEND
ALPD
SMUB
NUMBER OF PIVOTS
10^4
10^3
10^2
10
1
0
20
40
60
80
100
% OVERLAP

Figure (4.6.5)

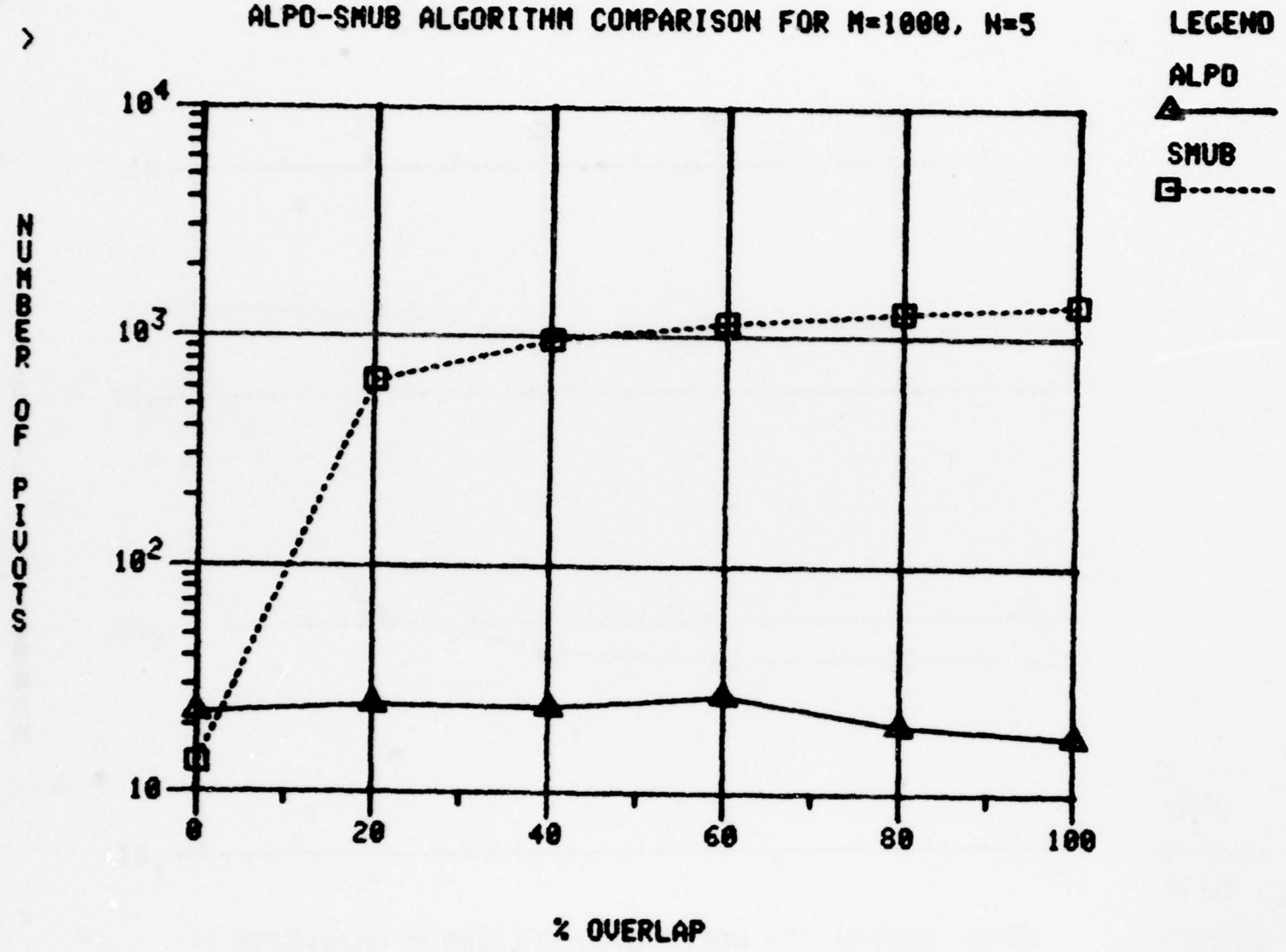

ALPD-SMUB ALGORITHM COMPARISON FOR M=1000, N=5
LEGEND
ALPD
SMUB
NUMBER OF PIVOTS
10^4
10^3
10^2
10
0
20
40
60
80
100
% OVERLAP

Figure (4.6.6)

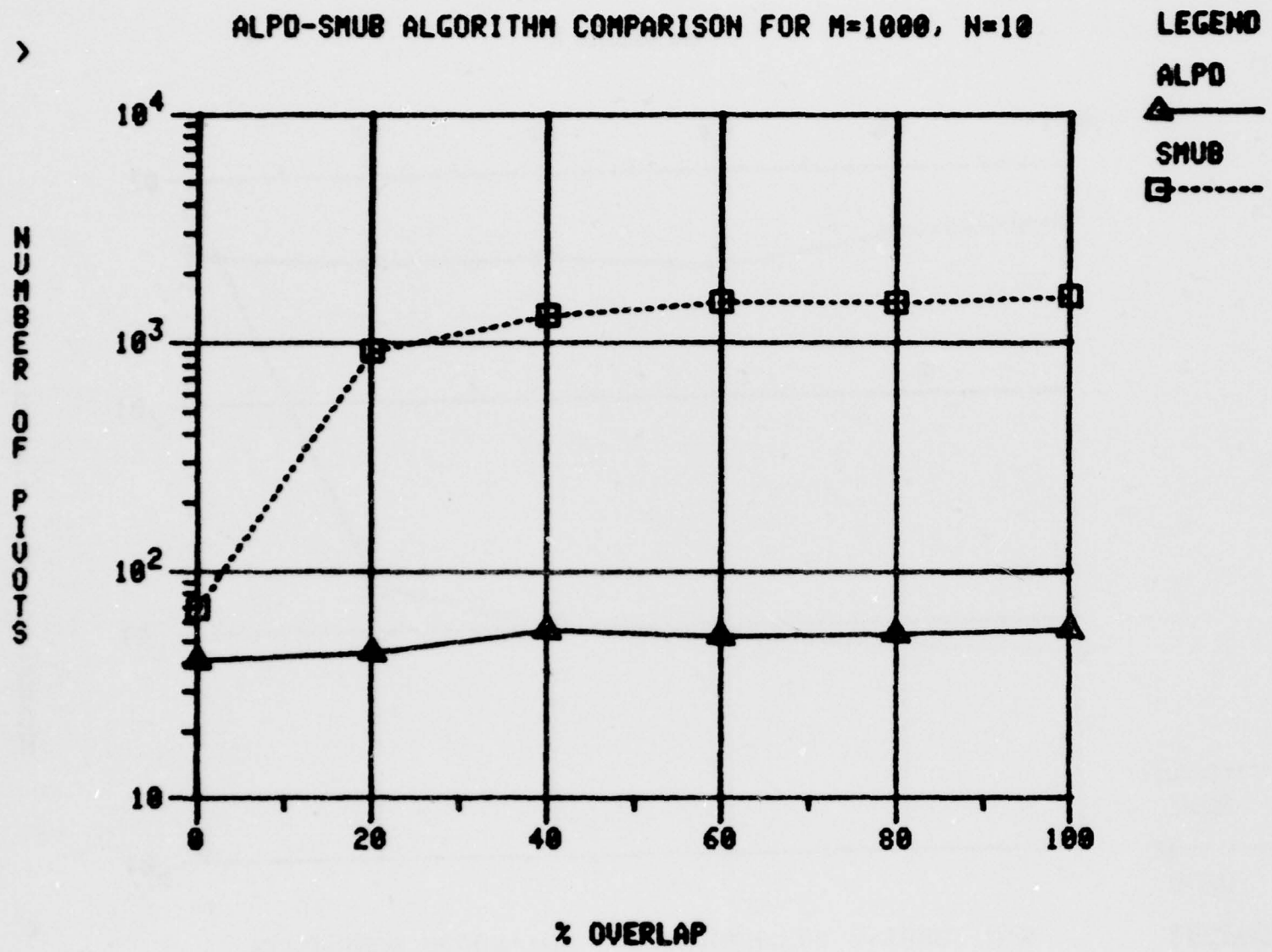

ALPO-SMUB ALGORITHM COMPARISON FOR M=1000, N=10
LEGEND
ALPO
SMUB
NUMBER OF PIVOTS
10^4
10^3
10^2
10
0
20
40
60
80
100
% OVERLAP

Figure (4.6.7)

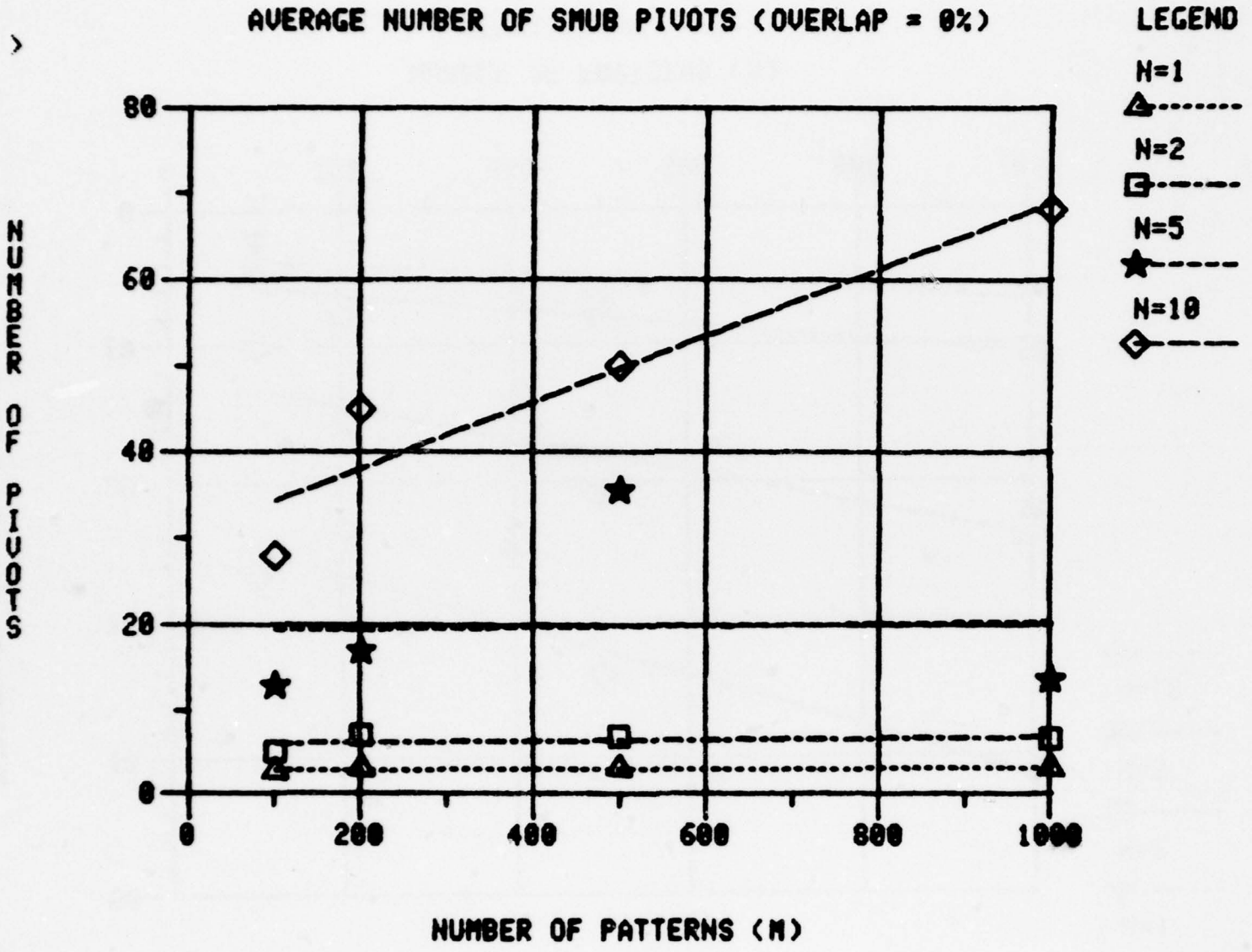

AVERAGE NUMBER OF SMUB PIVOTS (OVERLAP = 0%)
LEGEND
N=1
N=2
N=5
N=10
NUMBER OF PIVOTS
80
60
40
20
0
0
200
400
600
800
1000
NUMBER OF PATTERNS (M)

Figure (4.6.8)

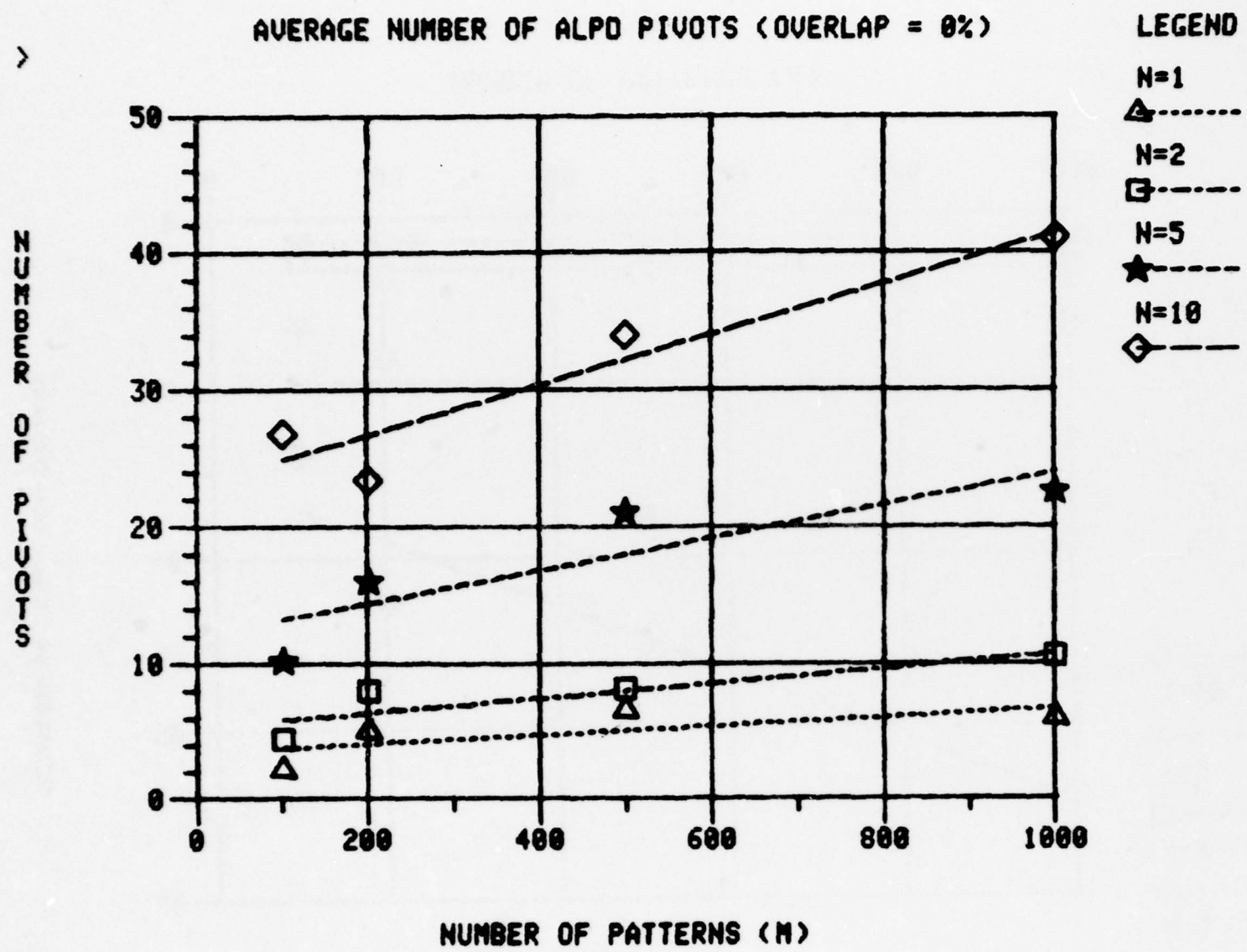

AVERAGE NUMBER OF ALPD PIVOTS (OVERLAP = 0%)
LEGEND
N=1
N=2
N=5
N=10
NUMBER OF PIVOTS
0
10
20
30
40
50
0
200
400
600
800
1000
NUMBER OF PATTERNS (M)

Figure (4.6.9)

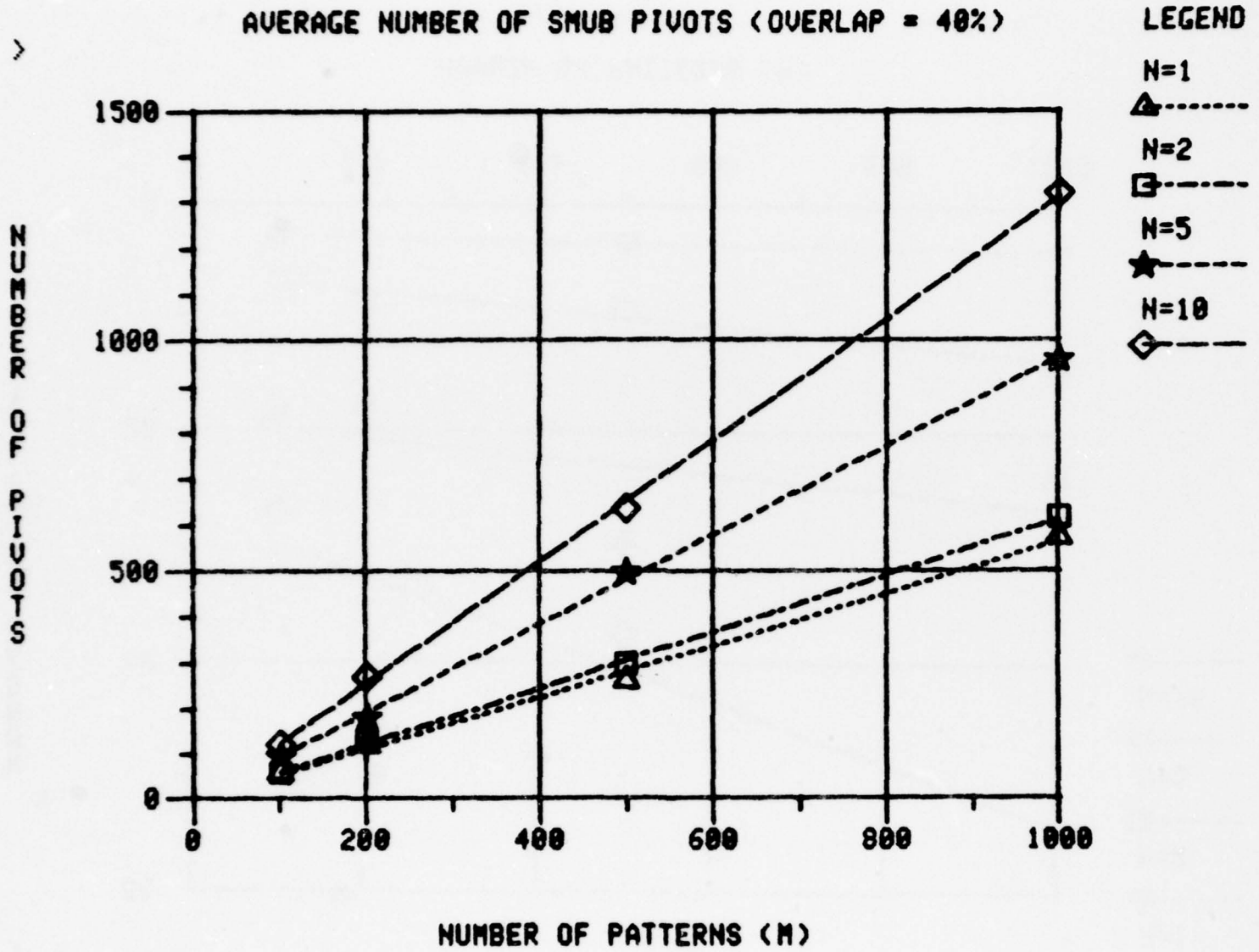

AVERAGE NUMBER OF SMUB PIVOTS (OVERLAP = 40%)
LEGEND
N=1
N=2
N=5
N=10
NUMBER OF PIVOTS
1500
1000
500
0
0
200
400
600
800
1000
NUMBER OF PATTERNS (M)

Figure (4.6.10)

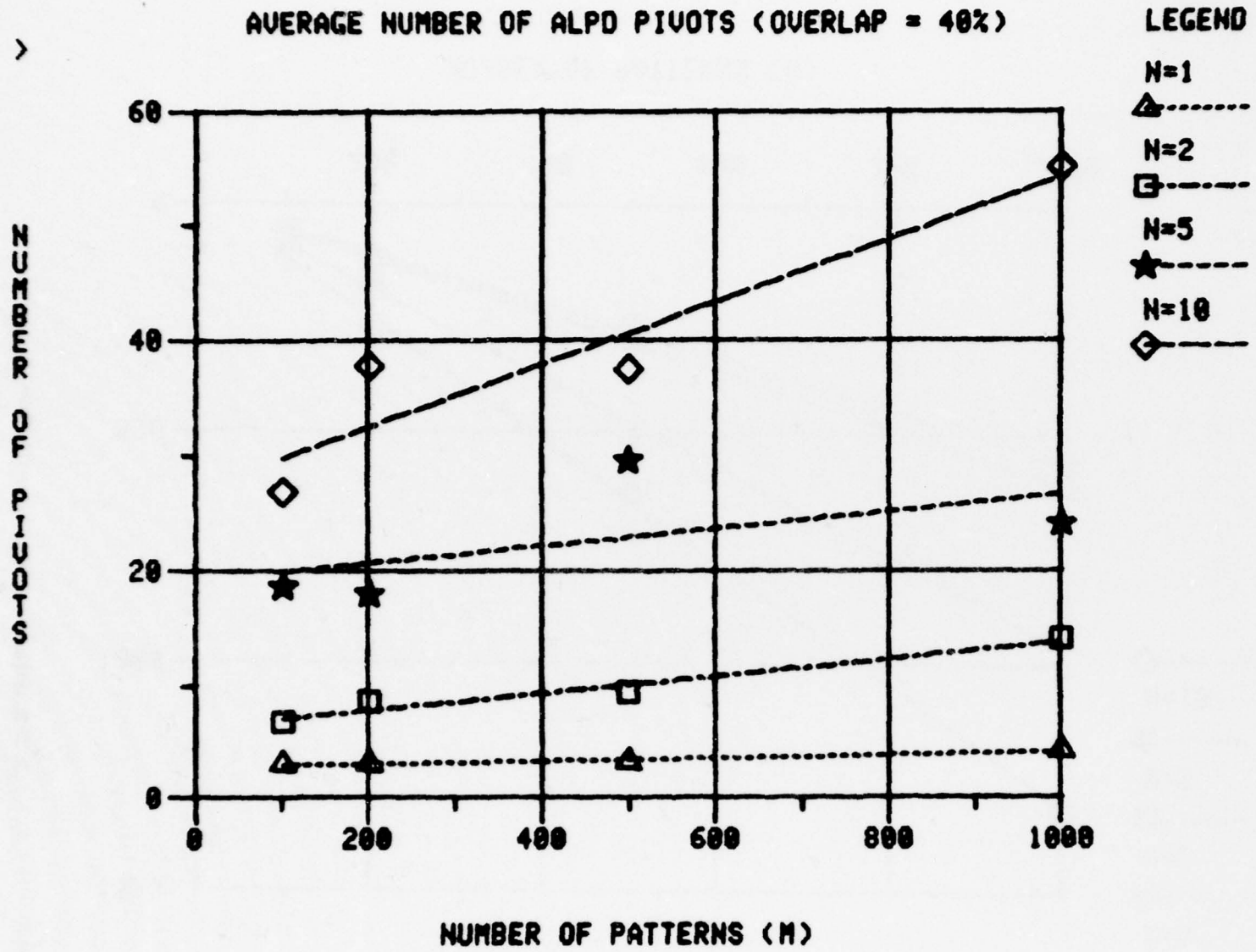

AVERAGE NUMBER OF ALPD PIVOTS (OVERLAP = 40%)
LEGEND
N=1
N=2
N=5
N=10
NUMBER OF PIVOTS
60
40
20
0
0
200
400
600
800
1000
NUMBER OF PATTERNS (M)

Figure (4.6.11)

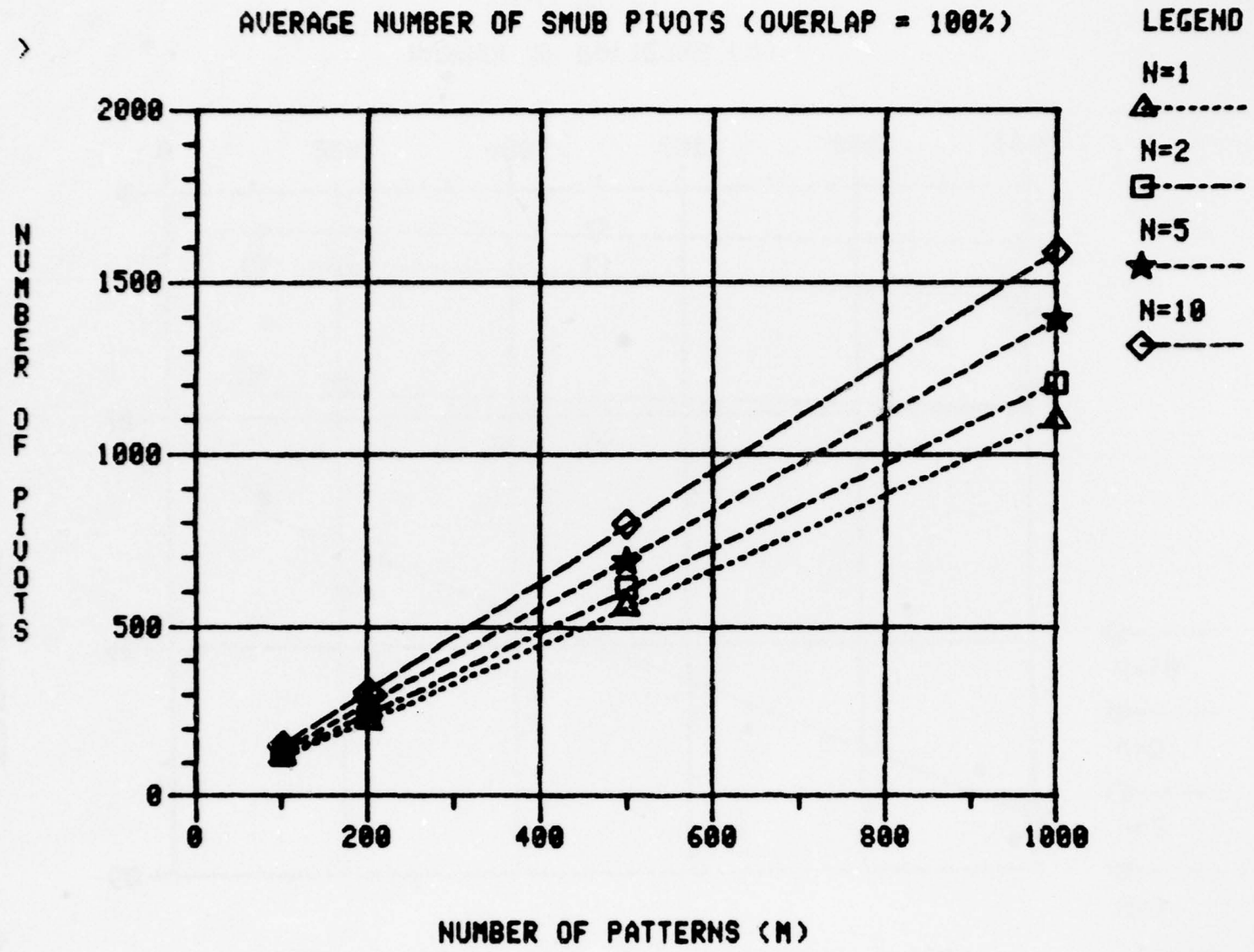

AVERAGE NUMBER OF SMUB PIVOTS (OVERLAP = 100%)
LEGEND
N=1
N=2
N=5
N=10
NUMBER OF PIVOTS
2000
1500
1000
500
0
0
200
400
600
800
1000
NUMBER OF PATTERNS (M)

Figure (4.6.12)

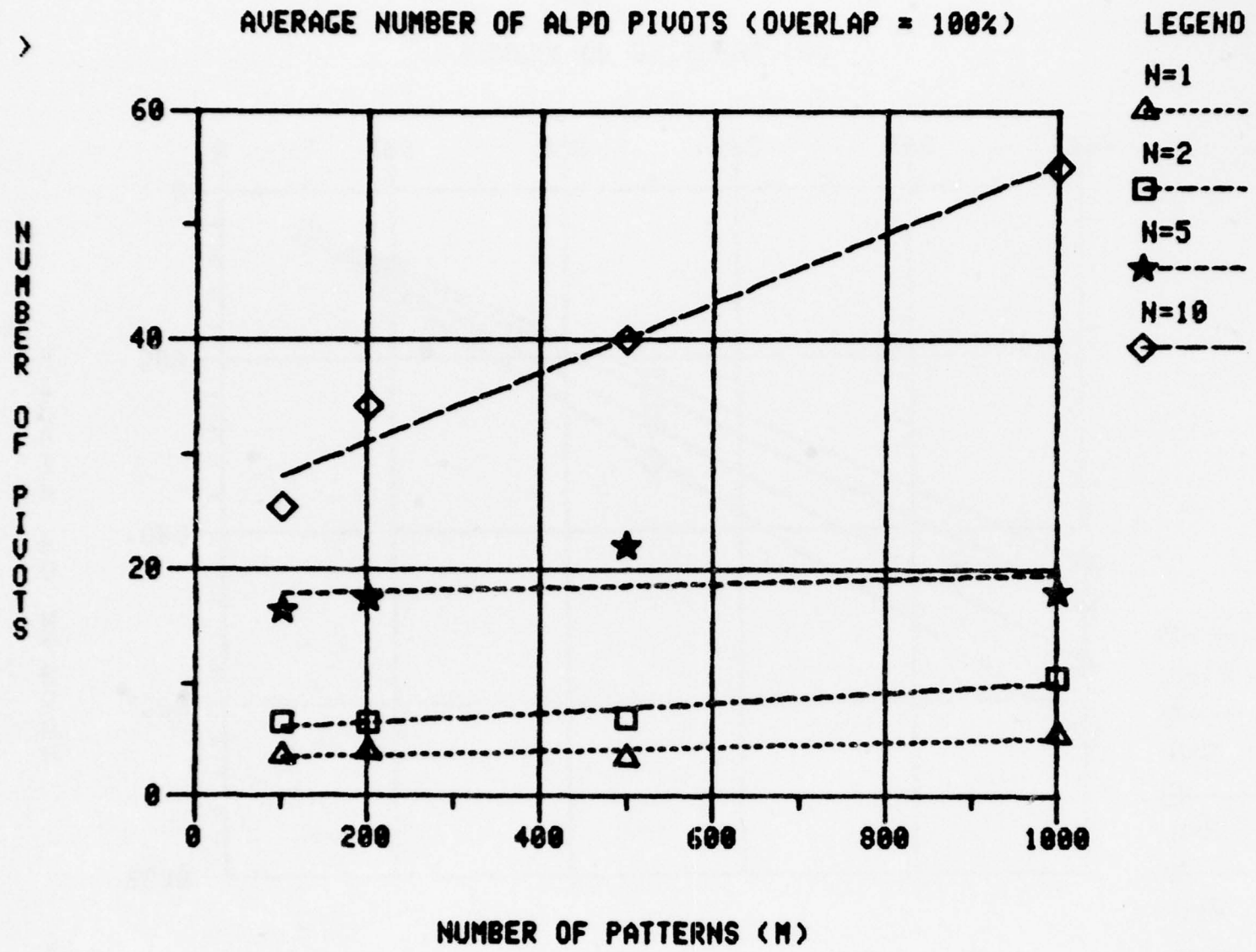

AVERAGE NUMBER OF ALPD PIVOTS (OVERLAP = 100%)
LEGEND
N=1
N=2
N=5
N=10
NUMBER OF PIVOTS
60
40
20
0
0
200
400
600
800
1000
NUMBER OF PATTERNS (M)

Figure (4.6.13)

CHAPTER 5

PIECEWISE LINEAR DISCRIMINANTS

5.1. Piecewise Linear Discriminants

A direct generalization of the linear discriminant is the piecewise linear discriminant. Piecewise linear functions $f: \mathbb{R}^n \rightarrow \mathbb{R}^1$ can be defined recursively as follows (Chang [22]):

Definition (5.1.1). Piecewise Linear Function

1. Any linear function $f(x) = w \cdot x - \theta$ is piecewise linear.
2. If $f_1(x)$, $f_2(x)$ are piecewise linear, then so are

$$\text{(5.1.1)} \qquad \text{and} \qquad \begin{aligned} f(x) &= \max\{f_1(x),\ f_2(x)\} \\ g(x) &= \min\{f_1(x),\ f_2(x)\} \end{aligned}$$

3. No other functions are piecewise linear.

Piecewise linear functions of arbitrary complexity can be constructed by repeated use of rule 2 in (5.1.1). With "$\vee$" and "$\wedge$" representing the maximum and minimum operators respectively, the following identities are useful for manipulating expressions involving piecewise linear functions.

$$
\begin{aligned}
&\text{a)} \quad f_1 \wedge (f_2 \vee f_3) = (f_1 \wedge f_2) \vee (f_1 \wedge f_3) \\
&\text{b)} \quad f_1 \vee (f_2 \wedge f_3) = (f_1 \vee f_2) \wedge (f_1 \vee f_3) \\
&\text{c)} \quad -(f_1 \vee f_2) = -f_1 \wedge -f_2 \\
&\text{d)} \quad -(f_1 \wedge f_2) = -f_1 \vee -f_2
\end{aligned} \tag{5.1.2}
$$

By repeated use of the distributive property a) and the associativity and commutativity of the minimum and maximum operators, any piecewise linear function f can be written in <u>disjunctive normal form</u>

$$
f = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{n_i} f_{ij} \right) , \tag{5.1.3}
$$

where each f_{ij} is linear.

This representation has the following geometrical interpretation Let $\mathcal{R} = \{x : f(x) \geq 0\}$. Then $\mathcal{R} = \bigcup_{i=1}^{m} \mathcal{R}_i$, where $\mathcal{R}_i$ is the polyhedral convex set defined by the linear inequality system

$$
\begin{aligned}
f_{i1}(x) &\geq 0 \\
&\vdots \\
f_{in_i}(x) &\geq 0
\end{aligned} \tag{5.1.4}
$$

Thus each concave function $f_i = \left(\bigwedge_{j=1}^{n_i} f_{ij} \right)$ in (5.1.3) isolates a convex region $\mathcal{R}_i$ whose boundaries are defined by the hyperplanes $f_{i1}(x) = 0$, ..., $f_{in_i}(x) = 0$. In a two-class pattern classification problem with pattern sets $\mathcal{S}_1$ and $\mathcal{S}_2$, if each region contains patterns only from $\mathcal{S}_1$ and together the regions contain all of the patterns in $\mathcal{S}_1$, then $f = \bigvee_{i=1}^{m} f_i$ is a piecewise linear discriminant that separates

$\mathcal{S}_1$ from $\mathcal{S}_2$. The situation is illustrated in Fig. (5.1.5).

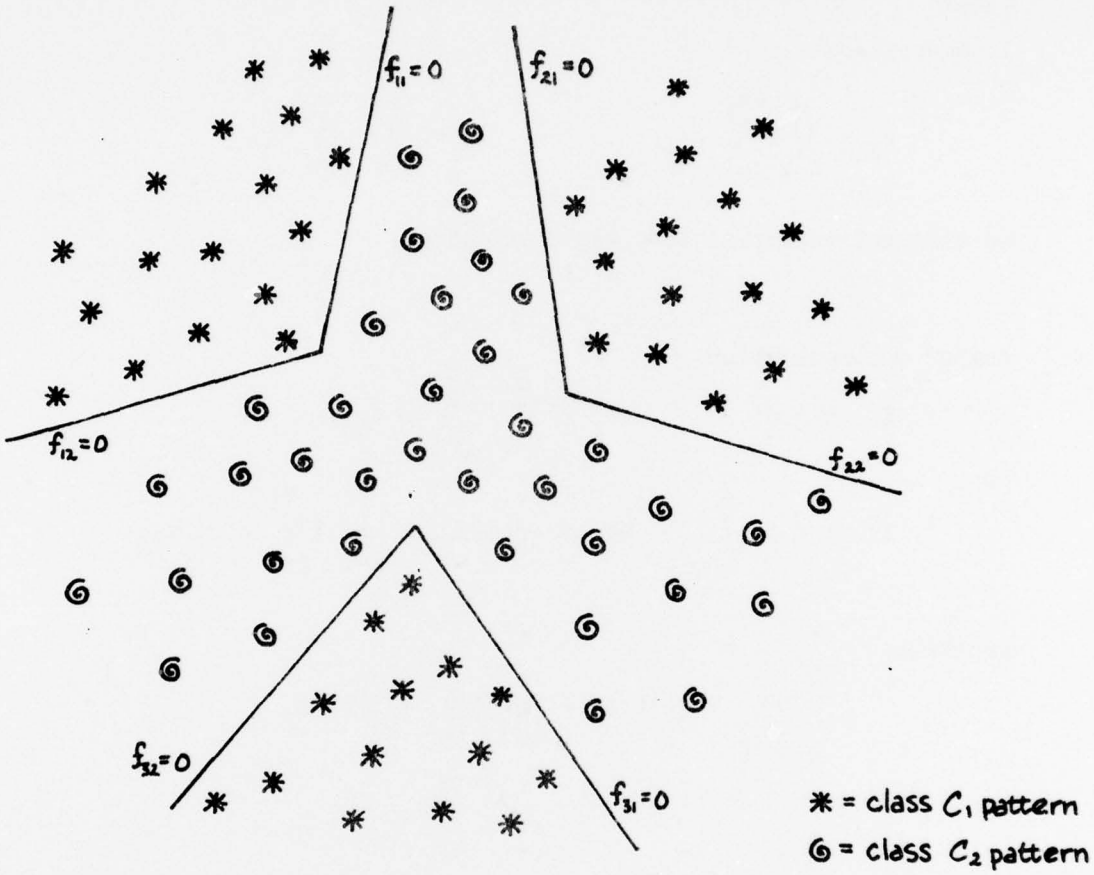


Figure (5.1.5). The Piecewise Linear Function

$$(f_{11} \wedge f_{12}) \vee (f_{21} \wedge f_{22}) \vee (f_{31} \wedge f_{32})$$

Separates the Two Pattern Classes.

The disjunctive normal form representation (5.1.3) can be used to show that piecewise linearity is preserved under the operations of addition and scalar multiplication.

PROPOSITION (5.1.6). If f_1 and f_2 are piecewise linear functions then $\alpha f_1 + \beta f_2$ is piecewise linear for all real constants α, β.

Proof. It is sufficient to show that αf_1 and $f_1 + f_2$ are piecewise linear. Let

$$f_1 = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{n_i} \ell_{ij} \right), \qquad f_2 = \bigvee_{p=1}^{r} \left(\bigwedge_{q=1}^{s_p} m_{pq} \right)$$

be disjunctive normal form representations.

Scalar multiplication:

$$\text{If } \alpha \geq 0, \qquad \alpha f_1 = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{n_i} \alpha \ell_{ij} \right)$$

$$\text{If } \alpha < 0, \qquad \alpha f_1 = -(|a| f_1) = \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{n_i} -|\alpha| \ell_{ij} \right)$$

Addition:

$$f_1 + f_2 = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{n_i} \ell_{ij} \right) + f_2$$

$$= \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{n_i} (\ell_{ij} + f_2) \right)$$

But $\ell_{ij} + f_2 = \bigvee_{p=1}^{r} \left(\bigwedge_{q=1}^{s_p} (\ell_{ij} + m_{pq}) \right)$, which is piecewise linear. □

5.2. Some Examples.

Many pattern classification schemes implicitly use piecewise linear discriminants. An example is the minimum distance classifier, also known as the "nearest neighbor" rule.

Let $\mathcal{P} = \{p_1, \ldots, p_j\}$, $\mathcal{Q} = \{q_1, \ldots, q_k\}$ be sets of prototype patterns representing classes C_1 and C_2, respectively. Let

$$(5.2.1) \qquad \begin{aligned} d_1(x) &= \min_{i=1,\ldots,j} \{\|x - p_i\|_2\} \\ d_2(x) &= \min_{i=1,\ldots,k} \{\|x - q_i\|_2\} . \end{aligned}$$

A minimum distance classifier is defined to be a classification procedure that implements the discriminant function

$$(5.2.2) \qquad f(x) = d_2^2(x) - d_1^2(x) .$$

Thus a pattern x is classified into the class of the nearest prototype pattern as measured by Euclidean distance. This discriminant is piecewise linear by Proposition (5.1.6) since it can be written in the form

$$(5.2.3) \qquad f(x) = \min_{i=1,\ldots,k} \{-2q_i \cdot x + \|q_i\|_2^2\} - \min_{i=1,\ldots,j} \{-2p_i \cdot x + \|p_i\|_2^2\} .$$

Minimum distance classifiers are particularly effective in situations where the patterns in each class cluster into isolated subclasses. If the clusters are sufficiently far apart, then a single prototype pattern selected from each subclass and included in the appropriate set $\mathcal{P}$ or $\mathcal{Q}$ will insure good performance of the discriminant on that subclass. Such multimodal behavior is sufficiently common that the problem of clustering multidimensional data has received much attention (e.g. [5], Ch. 6).

Even in the absence of such clustering behavior, a minimum distance discriminant can always be found that separates two finite, disjoint pattern sets $\mathcal{S}_1$ and $\mathcal{S}_2$. This follows immediately from the choice $\mathcal{P} = \mathcal{S}_1$, $\mathcal{Q} = \mathcal{S}_2$. When a minimum distance classifer uses prototype sets consisting of large numbers of known sample patterns from classes C_1 and C_2, respectively, the terminology "nearest neighbor rule" is often used to describe the classification procedure. Cover and Hart [23] show that if the known sample patterns are drawn from the same mixture distribution that produces the test patterns, the asymptotic error rate on new patterns as the number of known samples increases without bound is less than twice the error rate of the Bayes discriminant. However, this performance is achieved at the very considerable price of a large data storage requirement for the list of prototype patterns and the computational effort required to identify the nearest known sample to a test pattern.

Another example of a piecewise linear discriminant is found in the layered network of threshold logic units discussed in Section 2.2. Nilsson [24] shows that if there are k TLU's in the first layer, then a layered machine implements a discriminant of the form

$$f(x) = \max_{i=1,\ldots,j} \{f_i(x)\} - \max_{i=j+1,\ldots,2^k} \{f_i(x)\} \tag{5.2.4}$$

where each $f_i(x)$ is linear.

In the next section piecewise linear discriminants of the form

$$f(x) = \bigvee_{i=1}^{k} f_i(x)$$

where each $f_i(x)$ is linear are considered. Necessary and sufficient conditions for the existence of a discriminant of this type that separates two given finite pattern sets are developed. General applicability of this discriminant to arbitrary finite pattern sets is then demonstrated by use of a class of pattern space transformations.

5.3. Convex Separability

Let $\mathcal{S}_1$, $\mathcal{S}_2$ be subsets of $\mathcal{R}$ where $\mathcal{R}$ is a convex subset of $\mathbb{R}^n$.

Definition (5.3.1). $\mathcal{S}_1$ is convex separable from $\mathcal{S}_2$ if there exists a continuous convex function $f:\mathcal{R} \to \mathbb{R}^1$ such that

$$f(x) > 0 \qquad \forall\, x \in \mathcal{S}_1$$

$$f(x) < 0 \qquad \forall\, x \in \mathcal{S}_2$$

PROPOSITION (5.3.2). Let $\mathcal{S}_1 = \{x_1, \ldots, x_k\}$ be a finite point set and let $\mathcal{S}_2$ be any subset of $\mathbb{R}^n$. If $\mathcal{S}_1$ is convex separable from $\mathcal{S}_2$, then there exists a convex piecewise linear separating function $f(x)$.

Proof. Let $g(x)$ be a continuous convex separating function and let $\mathcal{R} = \{x:g(x) < 0\}$. $\mathcal{R}$ is an open convex region whose closure $\bar{\mathcal{R}}$ contains $\mathcal{S}_2$ as a proper subset and does not intersect $\mathcal{S}_1$. Thus by the separating hyperplane theorem, for each $x_i \in \mathcal{S}_1$ there exists a hyperplane

$w_i \cdot x = \theta_i$ which separates x_i from $\bar{\mathcal{R}}$, i.e. $f_i(x) = w_i \cdot x - \theta_i$ is positive for $x = x_i$ and negative for all $x \in \bar{\mathcal{R}}$. Then $f(x) = \bigvee_{i=1}^{k} f_i(x)$ is a convex piecewise linear function that separates $\mathcal{S}_1$ from $\mathcal{S}_2$. □

Proposition (5.3.2) can be used to prove the following geometric criterion for convex separability of finite pattern sets. Let $C(\mathcal{S})$ denote the convex hull of $\mathcal{S}$.

PROPOSITION (5.3.3). Let $\mathcal{S}_1$, $\mathcal{S}_2$ be finite, disjoint pattern sets. Then $\mathcal{S}_1$ is convex separable from $\mathcal{S}_2$ iff $\mathcal{S}_1 \cap C(\mathcal{S}_2) = \emptyset$.

Proof. $C(\mathcal{S}_2)$ is a closed convex set. If $\mathcal{S}_1 \cap C(\mathcal{S}_2) = \emptyset$, then a convex piecewise linear separating function can be constructed as in the proof of Proposition (5.3.2). Conversely, if a continuous convex function f separates $\mathcal{S}_1$ from $\mathcal{S}_2$, f is strictly positive on $\mathcal{S}_1$ and strictly negative on $\mathcal{S}_2$. By convexity, f is also strictly negative on $C(\mathcal{S}_2)$. Hence $\mathcal{S}_1 \cap C(\mathcal{S}_2) = \emptyset$. □

Figure (5.3.4) demonstrates that convex separability is not a symmetric relation between $\mathcal{S}_1$ and $\mathcal{S}_2$. Here $\mathcal{S}_1 \cap C(\mathcal{S}_2) = \emptyset$, but $C(\mathcal{S}_1) \cap \mathcal{S}_2 \neq \emptyset$.

Clearly not all disjoint pattern sets are convex separable. However, the following sufficient condition for convex separability motivates a class of coordinate transformations that render all finite disjoint pattern sets convex separable in the transformed space.

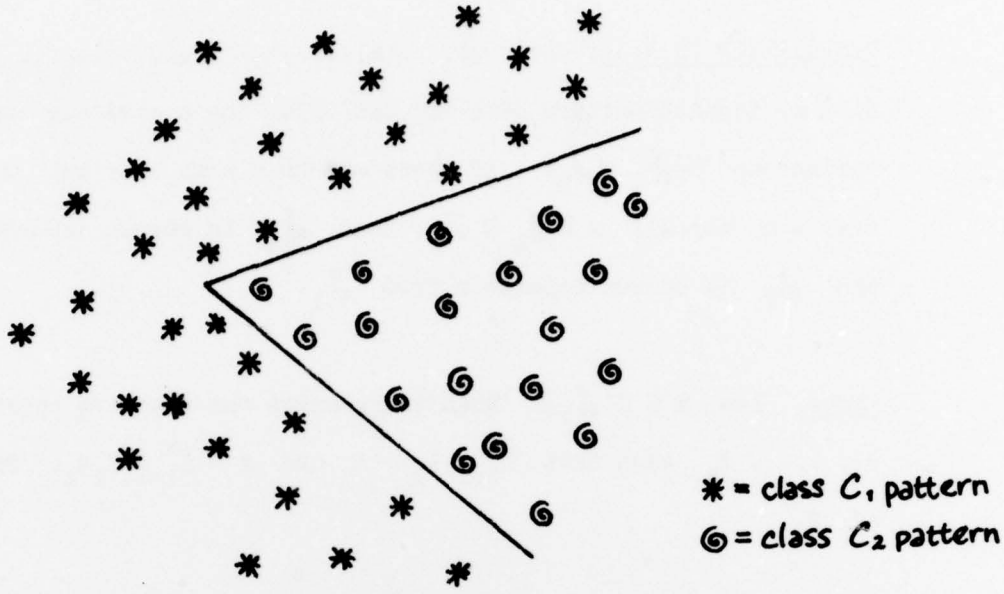


Figure (5.3.4). Class C_1 Patterns Are Convex Separable from Class C_2 Patterns But Not Conversely.

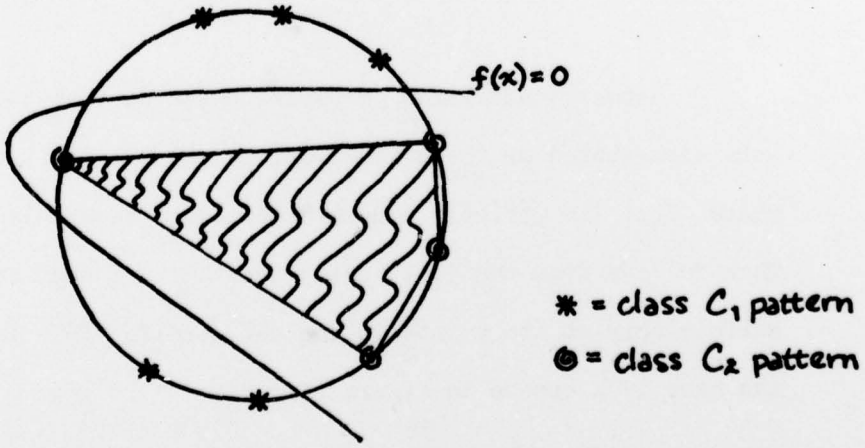


Figure (5.3.6). Finite Disjoint Pattern Sets on the Surface of a Sphere Are Always Convex Separable. The Convex Hull of the Class C_2 Patterns, Except for the Patterns Themselves Lies Inside the Sphere and Thus Cannot Contain Any Class C_1 Patterns.

PROPOSITION (5.3.5). Let $\mathcal{S}_1 = \{x_1,\ldots,x_k\}$, $\mathcal{S}_2 = \{x_{k+1},\ldots,x_m\}$ be finite disjoint pattern sets and let $f(x)$ be a strictly convex function defined on $C(\mathcal{S}_1 \cup \mathcal{S}_2)$. If there exists a real constant α such that $f(x) = \alpha$ for all $x \in \mathcal{S}_1 \cup \mathcal{S}_2$, then $\mathcal{S}_1$ is convex separable from $\mathcal{S}_2$ and $\mathcal{S}_2$ is convex separable from $\mathcal{S}_1$.

Proof. Let $x \in C(\mathcal{S}_1)$. Then there exist non-negative constants $\lambda_1, \ldots, \lambda_k$ such that $\sum_{i=1}^{k} \lambda_i = 1$ and $x = \sum_{i=1}^{k} \lambda_i x_i$. By convexity of f,

$$f(x) \leq \sum_{i=1}^{k} \lambda_i f(x_i) = \alpha .$$

By strict convexity of g, $f(x) < \alpha$ if x is not an extreme point of $C(\mathcal{S}_1)$, i.e. if $x \notin \mathcal{S}_1$. Since $\mathcal{S}_1$ and $\mathcal{S}_2$ are disjoint and $f(x) = \alpha$ for all $x \in \mathcal{S}_2$, $C(\mathcal{S}_1) \cap \mathcal{S}_2 = \emptyset$. By inverting the roles of $\mathcal{S}_1$ and $\mathcal{S}_2$, it follows also that $\mathcal{S}_1 \cap C(\mathcal{S}_2) = \emptyset$. □

Geometrically this proposition states that two disjoint pattern sets distributed on the surface defined by the equation $f(x) = \alpha$, where $f(x)$ is strictly convex, are convex separable from each other. This follows from the fact that convex hull of each set intersects the surface only at the points in the set itself. This is illustrated for the case of a sphere in Figure (5.3.6).

Example (5.3.7).

Disjoint binary pattern sets in $\mathbb{R}^n$, where each pattern component is either equal to + 1 or -1, are convex separable since they satisfy the hypotheses of Proposition (5.3.5) with

$$f(x) = \sum_{i=1}^{n} (x)_i^2 \quad \text{and} \quad \alpha = n . \qquad \square$$

For two general finite disjoint pattern sets $\mathcal{S}_1$, $\mathcal{S}_2$ in $\mathbb{R}^n$, it is possible to define a one-to-one mapping into sets $\mathcal{S}'_1$, $\mathcal{S}'_2$ in $\mathbb{R}^{n+1}$ such that $\mathcal{S}'_1$ and $\mathcal{S}'_2$ are convex separable. Let $g: \mathbb{R}^n \to \mathbb{R}^1$ be a strictly convex function defined on $C(\mathcal{S}_1 \cup \mathcal{S}_2)$. Let $h: \mathbb{R}^1 \to \mathbb{R}^1$ be a strictly convex function with an inverse $h^{-1}: \mathbb{R}^1 \to \mathbb{R}^1$ defined on $\{\alpha - g(x): x \in \mathcal{S}_1 \cup \mathcal{S}_2\}$ for some real constant α. For each $x \in \mathcal{S}_1 \cup \mathcal{S}_2$, define the transformed pattern $y \in \mathbb{R}^{n+1}$ by

$$y = (x, h^{-1}(\alpha - g(x))) \tag{5.3.8}$$

Let $\mathcal{S}'_1$, $\mathcal{S}'_2$ be the sets resulting from applying the transformation (5.3.8) to the patterns in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively.

PROPOSITION (5.3.9). The transformed pattern sets $\mathcal{S}'_1$, $\mathcal{S}'_2$ are convex separable from each other.

Proof. The function $f: \mathbb{R}^{n+1} \to \mathbb{R}^1$ defined by

$$f(x,\beta) = g(x) + h(\beta), \qquad x \in \mathbb{R}^n, \quad \alpha \in \mathbb{R}^1$$

is strictly convex. The set $\mathcal{S}_1'$ and $\mathcal{S}_2'$ are disjoint and if $y \in \mathcal{S}_1' \cup \mathcal{S}_2'$,

$$\begin{aligned} f(y) &= f(x,\ h^{-1}(\alpha - g(x)) \\ &= g(x) + h(h^{-1}(\alpha - g(x))) \\ &= \alpha \end{aligned}$$

and the result follows from Proposition (5.3.5). □

The sets $\mathcal{S}_1'$, $\mathcal{S}_2'$ are formed by mapping the patterns in $\mathcal{S}_1$ and $\mathcal{S}_2$ onto the surface $f(y) = \alpha$ in a one-higher dimensional space. The following two examples provide pattern space transformations that are valid for all finite, disjoint pattern sets in $\mathbb{R}^n$.

Example (5.3.10).

Let $g(x) = \sum_{i=1}^{n} (x)_i^2$, $h(\beta) = \beta^2$. Choose $\alpha = \max_{i=1,\ldots,k} \{g(x_i)\}$. Then

$$y = (x,\ \sqrt{\alpha - g(x)}\ )$$

is the desired pattern space transformation. In this example the n-dimension patterns in $\mathcal{S}_1 \cup \mathcal{S}_2$ are mapped onto the surface of the (n+1)-dimensional sphere of radius $\sqrt{\alpha}$ centered at the origin. □

Example (5.3.11).

This example works for any strictly convex function $g(x)$, e.g. $g(x) = xCx$ where C is an $n \times n$ positive definite matrix. Choose $h(\beta) = -\ell n(\beta)$. Then the desired pattern space transformation is

$$y = (x,\ e^{g(x)})$$

The strictly convex function

$$f(x,\beta) = g(x) - \ell n(\beta)$$

is equal to zero for all transformed patterns y. □

In the next section an algorithm is presented that constructs a convex piecewise linear discriminant by the method suggested in the proof of Proposition (5.3.2). An arbitrary pattern is chosen from $\mathcal{S}_1$ and then a linear discriminant separating this pattern from the entire set $\mathcal{S}_2$ is found as a solution to a constrained LPD problem. The problem is designed to encourage the separation of as many as possible additional patterns in $\mathcal{S}_1$ from $\mathcal{S}_2$ along with the chosen one. All patterns in $\mathcal{S}_1$ that are separated from $\mathcal{S}_2$ are then dropped from $\mathcal{S}_1$ and the process is repeated with new linear discriminants until $\mathcal{S}_1$ is empty.

5.4. An Algorithm for Convex Piecewise Linear Separation

Let $\mathcal{S}_1 = \{x_1, \ldots, x_\ell\}$, $\mathcal{S}_2 = \{x_{\ell+1}, \ldots, x_m\}$ be finite disjoint pattern sets such that $\mathcal{S}_1$ is known to be convex separable from $\mathcal{S}_2$ (e.g. the patterns are binary or have undergone the transformation described in Section 5.3). An algorithm is now presented that determines a convex piecewise linear separating function.

Let

k = iteration number

$\mathcal{S}_1^{(k)}$ = set of patterns in $\mathcal{S}_1$ not yet separated from $\mathcal{S}_2$ before the kth iteration.

$x^{(k)}$ = selected element of $\mathcal{S}_1^{(k)}$.

$A_1^{(k)}$ = signed augmented pattern matrix corresponding to $\mathcal{S}_1^{(k)} - \{x^{(k)}\}$

A_2 = signed augmented pattern matrix corresponding to $\mathcal{S}_2$.

$a^{(k)}$ = signed augmented pattern corresponding to $x^{(k)}$.

ALGORITHM (5.4.1).

Step 1. Set $k = 1$, $\mathcal{S}_1^{(1)} = \mathcal{S}_1$. Go to Step 1.

Step 2. Choose an arbitrary pattern $x^{(k)} \in \mathcal{S}_1^{(k)}$. Form the matrix $A_1^{(k)}$ and solve the constrained LPD problem

$$\begin{aligned} & \min\ e \cdot s \\ \text{s.t.}\quad & A_1^{(k)} u + Is \geq e \\ & a^{(k)} u \qquad \geq 1 \\ & A_2 u \qquad \geq e \\ & \qquad\quad s \geq 0 \\ & u = (w, \theta) \in \mathbb{R}^{n+1} \end{aligned} \tag{5.4.2}$$

Let $u^{(k)} = (w^{(k)}, \theta^{(k)})$ be an optimal solution to (5.4.2). Go to Step 3.

Step 3. Set $\mathcal{S}_1^{(k+1)} = \{x_i \in \mathcal{S}_1^{(k)}, w^{(k)} \cdot x_i - \theta^{(k)} \leq 0\}$. If $\mathcal{S}_1^{(k+1)}$ is empty, go to Step 4. Otherwise increment k by 1 and go to Step 2.

Step 4. Stop. Let k^* be the final value of k. Then the desired convex piecewise linear separating function is

$$f^*(x) = \bigvee_{i=1}^{k^*} (w^{(i)} \cdot x - \theta^{(i)}) .$$

Proof of Algorithm:

Since $\mathcal{S}_1$ is assumed convex separable from $\mathcal{S}_2$, each individual pattern in $\mathcal{S}_1$ is linearly separable from $\mathcal{S}_2$. Thus the inequality system

$$(5.4.3) \qquad \begin{aligned} a^{(k)} \cdot u &\geq 1 \\ A_2 u &\geq e \end{aligned}$$

is feasible and hence an optimal solution $u^{(k)}$ to (5.4.2) exists by Proposition (3.5.11). Since $w^{(k)} \cdot x^{(k)} - \theta^{(k)} \geq 1$, $\mathcal{S}_1^{(k+1)}$ is smaller than $\mathcal{S}_1^{(k)}$ by at least one element for all $k < k^*$. Thus the algorithm must terminate in at most ℓ iterations. For each $x_i \in \mathcal{S}_1$, there is at least one value of k such that $w^{(k)} \cdot x_i - \theta^{(k)} \geq 1$. Hence $f^*(x) > 0$ for all $x \in \mathcal{S}_1$. Also, since $A_2 u^{(k)} \geq e$ for $k = 1, \ldots, k^*$, $f^*(x) < 0$ for all $x \in \mathcal{S}_2$. □

The linear program (5.4.2) produces a hyperplane that minimizes the sum of the infeasibilities corresponding to remaining class C_1 patterns subject to the constraint that all class C_2 patterns and a specified C_1 pattern are on the 'correct' side of their respective margin planes. Hopefully this LPD form of the objective function

encourages the optimal hyperplane to separate other class $\mathcal{C}_1$ patterns in addition to the specified one at each iteration whenever possible. Toward this end it has been found that for several test problems of the overlapping hypercube type discussed in Section (4.6), replacement of the constraints $A_2u \geq e$ in (5.4.2) with $A_2u \geq \epsilon e$, where ϵ is a very small positive number, often reduces the total number of iterations required. In effect, this change eliminates the margin problem for the class $\mathcal{C}_2$ patterns and forces the optimal hyperplane to pass very close to the convex hull of $\mathcal{S}_2$. Thus for sufficiently small values of ϵ, the possibility of a class $\mathcal{C}_1$ pattern lying between the optimal hyperplane and this convex hull is eliminated. Numerical experience with this revised form of the algorithm suggests that when the selected class $\mathcal{C}_1$ pattern is part of a cluster of $\mathcal{C}_1$ patterns that are linearly separable from $\mathcal{S}_2$, all or nearly all of the cluster is separated by the optimal hyperplane. The following example illustrates this behavior.

Example (5.4.4)

The overlapping hypercube problem discussed in Section (4.6) was selected as a test case. A total of $m = 200$ patterns of dimension $n = 2$ were generated, half in each class. The two unit squares overlapped on an area of $\alpha = 0.20$. To introduce convex separability, the patterns were mapped onto the surface of a three dimensional sphere by the transformation given in Example (5.3.10). The resultant three-dimensional patterns were separated by a convex function generated by the revised version of algorithm (5.4.1). The constrained LPD problems were solved

by the ALPD algorithm after conversion to a weighted LPD format. The separation sequence is shown in Table (5.4.5). The problem required a total of 11 iterations for complete separation. The first iteration hyperplane succeeded in separating a large cluster of 78 class C_1 patterns, while subsequent hyperplanes separated either isolated patterns or small clusters. This behavior is consistent with the geometry of the problem. In the original pattern space $(\mathbb{R}^2)$, the 20% overlap factor implies that a large fraction of the class C_1 patterns should be linearly separable from $\mathcal{S}_2$. Since the mapping of the patterns onto the sphere in $\mathbb{R}^3$ leaves the first two coordinates intact, linear separability of these patterns is preserved. The remaining class C_1 patterns in $\mathbb{R}^2$ are uniformly distributed in or near the area of overlap. Thus the transformed patterns in $\mathbb{R}^3$ from the overlap area in $\mathbb{R}^2$ are expected to show little tendency to cluster by class with only small linearly separable clusters of class C_1 patterns being formed by chance. □

Iteration Number	Remaining Class C_1 Patterns	Number of Class C_1 Patterns Separated
1	100	78
2	22	3
3	19	1
4	18	6
5	12	1
6	11	2
7	9	1
8	8	2
9	6	1
10	5	3
11	2	2

Table (5.4.5). Separation sequence of convex separation algorithm in Example (5.4.4)

REFERENCES

[1] McCulloch, W. S. and W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bulletin of Math. Biophysics, 5, pp. 115-133 (1943).

[2] Rosenblatt, Frank, Principles of Neurodynamics, Sparton Books, New York (1962).

[3] Minsky, M. and S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press, Cambridge, Mass. (1969).

[4] Novikoff, A. B. J. "On convergence proofs for perceptrons," Proc. Symp. on Math. Theory of Automata, pp. 615-622, Polytechnic Institute of Brooklyn, Brooklyn, N.Y. (1962).

[5] Duda, R.O. and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley, New York (1973).

[6] Grinold, R. C., "Mathematical programming methods of pattern classification," Management Science, 19 (Theory series), pp. 272-289 (November 1972).

[7] Mangasarian, O. L., Linear and non-linear separation of patterns by linear programming," Operations Research, 13, pp. 444-452 (May-June 1965).

[8] Ibaraki, T. and S. Maroga, "Adaptive linear classifier by linear programming," IEEE Transactions on System Science and Cybernetics, SSC-6, pp. 53-62 (January, 1970).

[9] Rosen, J. B., "Pattern separation by convex programming," Journal of Mathematics and Application, 10, pp. 123-134 (1965).

[10] Luenberger, D. G., Optimization by Vector Space Methods, John Wiley, New York (1969).

[11] Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton (1963).

[12] Bowley, A. L., F. Y. Edgeworth's Contributions to Mathematical Statistics, Royal Statistical Socieity, London (1928).

[13] Draper, N. R., and H. Smith, Applied Regression Analysis, John Wiley, New York, 1966.

[14] Rhodes, E. C., "Reducing observations by the method of minimum deviations," Philosophical Magazine, 7th series, pp. 974-989.

[15] Singleton, R. R., "A method for minimizing the sum of absolute values of deviations," Ann. Math. Stat., 11, pp. 301-310 (1930).

[16] Charnes, A., W. W. Cooper and R. D. Ferguson, "Optimal estimation of executive compensation by linear programming," Management Science, 1, pp. 138-151 (1955).

[17] Wagner, H. M., "Linear programming techniques for regression analysis," J. Amer. Stat. Assoc., 54, pp. 206-212 (1959).

[18] Davies, M., "Linear approximation using the criterion of least total deviations," J. Royal Stat. Soc., Series B, 29, pp. 101-109 (1967).

[19] Pierre, D. A., Optimization Theory with Applications, John Wiley, New York (1969).

[20] Patterson, J. D., and B. F. Womack, "An adaptive pattern classification system," IEEE Trans. Sys. Sci. Cyb., SSC-2, pp. 62-67 (August 1966).

[21] Smith, F. W., "Pattern classifier design by linear programming," IEEE Trans. Comp., C-17, pp. 367-372 (April 1968).

[22] Chang, C., "Pattern recognition by piecewise linear discriminants," IEEE Trans. Comp., C-22, pp. 859-862 (September 1973).

[23] Cover, T. M. and P. E. Hart, "Nearest neighbor pattern classification," IEEE Trans. Info. Theory, IT-13, pp. 21-27 (January, 1967).

[24] Nilsson, N. J., Learning Machines, McGraw-Hill, New York (1965).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 78-14	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MATHEMATICAL PROGRAMMING APPLICATIONS IN PATTERN RECOGNITION		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER 78-14
7. AUTHOR(s) Robert Hemstreet Leary		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0267
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-047-143
11. CONTROLLING OFFICE NAME AND ADDRESS Operations Research Program Department of the Navy, 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE August 1978 13. NUMBER OF PAGES 122
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Algorithm
Classification
Discriminant
Inequalities
Convexity

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

SEE ATTACHMENT

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

MATHEMATICAL PROGRAMMING APPLICATIONS IN PATTERN RECOGNITION

Robert H. Leary, 78-14

Problems in pattern recognition are treated by the methods of mathematical programming. In particular the two-class pattern classification model with decision rules based on discriminant functions is considered with emphasis on mathematical programs that determine linear and piecewise linear discriminants.

For linearly separable pattern sets separating hyperplane can be determined by solving a system of linear inequalities. This system serves as the constraint set for a class of mathematical programs that define separating linear discriminants exhibiting maximum tolerance to pattern noise. Specific cases that can be modelled as linear and quadratic programs are discussed and a reliability interpretation of the objective criterion is given.

Application of linear discriminants to the linearly inseparable case leads to consideration of solution concepts for possible infeasible linear inequality systems. The Least Positive Deviations (LPD) solution to the general system $Ax \geq b$, where A is a $(m \times n)$ matrix with $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, is defined by a Phase I linear programming model. An equivalent unconstrained minimization problem with a piecewise linear objective serves as the basis for the development of the Accelerated Least Positive Deviations (ALPD) algorithm for the solution of the model. The algorithm is shown to be implementable by a sequence of pivot operations of the same type as employed by the simplex method with upper bounds applied to the dual of the Phase I problem but with a novel pivot selection rule and without regard to the upper bounds. At each iteration the pivot selection is determined by the solution to an unconstrained minimization of a piecewise linear function of a one-dimensional variable. Like the simplex method, the ALPD algorithm converges in a finite number of iterations to an optimal solution. A direct comparison of the relative efficiencies of the simplex and ALPD algorithms can be made in terms of the number of basis changes required to reach optimality from the same arbitrary initial basis. Results of an extensive series of numerical tests are reported which indicate a large ALPD advantage for linearly inseparable classification problems. The advantage appears to increase with the aspect ratio (m/n) of the matrix A and the degree of infeasibility of the underlying inequality system.

The LPD problem is generalized to the weighted and constrained weighted least deviations problems, which are shown to be directly solvable by the ALPD algorithm. Examples of such problems are presented from linear estimation and control theory. The general linear programming problem is also formulated as a constrained weighted least deviations model. Properties of LPD and related models are explored for classification problems and an asymptotic LPD discriminant characterization is obtained.

The LPD methodology is extended to piecewise linear discriminants. A class of pattern space transformations is defined that renders any pair of finite disjoint pattern sets separable by a convex piecewise linear function. An algorithm is presented that constructs such a function through the solution of a sequence of constrained weighted least deviations problems. Results of a numerical test problem are presented.