THE EFFECTIVENESS OF FEATURES IN PATTERN RECOGNITION

by

SIDDHESWAR RAY

A Thesis Submitted for the Degree of

Doctor of Philosophy

of the University of London

Department of Electrical Engineering

Imperial College of Science and Technology

University of London

November 1984

To my
Dada and Baudi

ABSTRACT

Feature evaluation criteria are investigated in the context of pattern recognition. This investigation is composed of three broad parts: critical examination of the existing methods, proposition of new methods based on the Mahalanobis distance, and empirical study of existing and proposed methods.

Various two-class and multiclass probabilistic criteria, most of which originate from the concepts of overlap and/or distance between classes, are examined for their comparative assessment as measures of feature effectiveness. Bayesian error probability being an optimum measure of the performance of a pattern recognition system, the different methods are judged depending on their relationship with this error probability. The two-class measures considered include the Bhattacharyya coefficient, the Matusita distance, the divergence function, the Kolmogorov variational distance, the generalized separability measure of Lissack and Fu and the Mahalanobis distance. The multiclass measures include Matusita's measure of affinity, Shannon's conditional entropy, the Bayesian distance of Devijver, the conditional quadratic entropy of Vajda and Minkowski's measures of nonuniformity. Apart from these direct multiclass measures investigation is also made of the indirect multiclass generalizations of the aforesaid two-class measures obtained by averaging them over different class-pairs. Since none of the available measures, except the two-class Kolmogorov variational distance, has exact relationship with the Bayesian error probability, they are judged with reference to their lower and upper error probability bounds.

In contrast to the other feature evaluation methods mentioned above, the Mahalanobis distance does not require information about the probability structure. On the other hand, it shares with them the property of providing distribution-free probability of error upper bound. On account of this and other aspects the Mahalanobis distance is proposed as a feature evaluation criterion. Transformations on

Mahalanobis distance are put forward which lead to the avoidance of certain difficulties that arise when using a two-class measure in a multiclass situation.

Finally the existing and the proposed methods are applied to the problem of the recognition of handwritten numerals. Feature set considered for evaluation purposes consists of 20 'normalized frequency' features and 81 'normalized characteristic loci' features. The suggested transformations on Mahalanobis distance are shown to improve the feature evaluation process.

ACKNOWLEDGEMENTS

The work reported in this thesis was supervised by Prof. L.F. Turner. I am grateful for his guidance and criticism, and express my appreciation for creating a free atmosphere for research which contributed significantly to this work. I gratefully acknowledge the financial support given by the Commonwealth Commission through a 'Commonwealth Scholarship', the study leave granted by the Indian Statistical Institute and the permission given by the Government of India to stay abroad.

I am grateful to my colleagues and friends at Imperial College for all the help and advice I received from them. In particular, I am indebted to Mrs. S. Cambell and Mr. G. Fernando for their help in digitization of handprinted data, and to Dr. I. Habbab for allowing me to use his graph plotting program and for many stimulating discussions that we had during the last few years.

I wish to extend my thanks and appreciation to my Bengalee friends Mr. S. Biswas, Dr. A. Chakravarty and Dr. D. Saha whose company was a constant source of pleasure to me and without whom my stay in London would have been far less enjoyable.

Finally, I wish to express my sincere gratitude to Dr. S. Chatterjee and Mrs. D. Chatterjee for kindly looking after and bearing with me during the final stages of my thesis preparation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Description of the Problem and Objectives of the Thesis

Pattern recognition can be viewed as a three-stage process involving pattern representation, feature selection and classification. Pattern representation is the process of transforming an input pattern into a form suitable for computer processing. The output of this process would be a set of measurements representing the input pattern. This set of measurements constitutes, what is often called, a 'measurement vector'. Feature selection is the process of reduction of dimensionality of the measurement vector. This dimensionality reduction is achieved either by discarding the redundant and less relevant measurements or by combining the original measurements to form a set of characteristics representing the input pattern or by both. These new characteristics are said to constitute a 'feature vector'. In the classification stage a decision as to which category the input pattern may belong to is then made based on the value of the feature vector.

Usually the elements of the measurement vector represent some physical properties of the input pattern whereas the elements of the feature vector are mathematical in nature. Therefore, in contrast to the process of pattern representation, feature selection and classification methodologies can be developed without being

constrained by their applications. This works as an encouragement for more research in these two areas.

The criterion for measuring the performance of a pattern recognition system depends on its ultimate objective. Minimization of cost of classification, maximization of classification accuracy, description and analysis of the input data for the purpose of their reproduction, and measurement of distance between the pattern classes are some of the objectives. For the purpose of the present study it is assumed that the underlying objective is to maximize the classification accuracy.

The feature selector and the classifier are jointly responsible for the performance of a pattern recognition system. Accuracy of one affects that of the other. Considerable success has already been achieved in the design of pattern classifier. Design of feature selector, i.e., selection of features to be used in a pattern recognition problem, is a comparatively difficult task. Though some efforts have been made in the past, a satisfactory answer to the problem of optimum feature selection is still not available. In the present study the concentration is on the methods of feature evaluation used for the purpose of selection of effective features.

Since the performance of a pattern recognition system depends on the classifier also, the assumption of Bayesian clssifier with (0,1) cost function seems to be ideal for comparing the effectiveness of various feature subsets. In this case the resultant Bayesian error probability (P_e, say) would indicate how well a subset of features can

describe the input patterns from the point of view of their recognition. An alternative view of the effectiveness of a set of features relates to the ease with which optimum or near optimum decision-making can be performed. This is an aspect that will not be considered in this study.

Although the Bayesian error probability is usually considered to be an optimum measure of the effectiveness of features, in most cases either a closed-form expression cannot be obtained for it or it is too difficult to compute numerically*. Owing to these difficulties various indirect methods of evaluation and selection of features have been developed during the past twenty five years. Most of these methods fall under two broad types of approaches. The first type consists of measures of feature effectiveness which are expressed in terms of the probability distributions characterizing the pattern classes and may, therefore, be termed as 'probabilistic measures'**. They include various information, distance and dependence measures. Though these criteria do not bear any exact relationship with P_e, they approximate it by providing lower and upper bounds to P_e.

---

* These problems will be discussed in detail in the introductory section of chapter 2.

** The term 'probabilistic measure' is used in this thesis in a general sense and should not be confused with the term 'probability measure' found in the Statistics literature.

The second type consists of mathematical mapping criteria which transform a feature vector*, into another feature vector of lower dimensionality. From the point of view of pattern recognition an ideal transformation would lead to optimum reduction in the number of features together with the minimization of Bayesian probability of error. Unfortunately, in most cases, the existing mapping criteria do not fulfil this optimality condition. In the next section a comparative analysis of the above two approaches will be made. At this point it may be mentioned that in the first approach different feature subsets are evaluated to decide which one of them to select whereas in the second approach the process of feature evaluation is not required and a mapping criterion leads to a smaller set of features straightway.

The purpose of the present study is to make a critical review of the existing probabilistic criteria of feature evaluation, the first of the two approaches mentioned above, and then to propose some new Mahalanobis distance-based criteria. The justification for concentrating on the Mahalanobis distance lies in its advantage over the approaches mentioned above. It is computationally less complex than the probabilistic criteria but provides probability of error bound which is lacking** in the mathematical techniques.

---

* Measurement vector may be considered to be the initial feature vector, thus generalizing the concept of a 'feature vector'.

** If the features are Gaussianly distributed then most of the mathematical techniques also provide bounds to the probability of error.

The probabilistic measures and the proposed Mahalanobis distance-based criteria are first examined on the basis of their P_e bounds. They are then applied to the problem of recognition of handprinted numerals as a means of comparing them experimentally. It may be mentioned here that the application area of handprinted numeral recognition is chosen simply for the purpose of comparison of different feature evaluation methods, and the recognition accuracy of handprinted numerals should not be considered as the goal of the present study. Two types of experimental comparison are made. Firstly, various feature evaluation methods are applied to arrange a set of features chosen for recognition of numerals and then rank correlations between pairs of feature orderings are computed. These correlation values give an idea about the conformity of the orderings. The criterion whose feature ordering has maximum rank correlation with the ordering by the Bayesian error criterion is expected to perform better than the other measures. Secondly, recognition experiments are performed based on the feature sets selected by various criteria. This is repeated for different feature set sizes. The resultant recognition accuracies will enable a comparison of the performances of various feature evaluation criteria with respect to one another for different feature set sizes.

1.2 A Brief Historical Background

With the advent of digital computers pattern recognition research began during the late 1950's. An account of the early works may be obtained from Unger's paper [1]. Soon there was a great demand for the development of pattern recognition methodology. As is evident from the survey papers of Wee [2], Nagy [3], and Ho and Agrawala [4], during 1960's the main focus of research was on designing the classifier. This is not at all surprising because, although all the three stages of representation, feature selection and classification are responsible for the performance of a pattern recognition system, classification is the stage in which the final decision-making takes place. Nevertheless, the problem of feature selection was not totally ignored during those days. As early as in 1962 Lewis produced a paper [5] which is considered to be pioneering in the field. He proposed the use of mutual information as a feature evaluation measure and gave theoretical justifications in its support. In his experiments an approximation of the above measure was used. Lewis's work was followed by Kamentsky and Liu [6], [7] who applied the mutual information criterion to evaluate various logic circuits for the purpose of designing multifont character recognition logics. Marill and Green [8] were probably the first to suggest the use of the divergence measure as a feature evaluation criterion. This measure was initially proposed by Jeffreys [9] and several of its properties were derived by Kullback [10]. The paper of Kailath [11] was another landmark in the history of development of feature evaluation methodology. In this paper he defined a new distance measure based on

the Bhattacharyya coefficient [12] and called it the Bhattacharyya distance. He compared the properties of the new measure with those of the divergence function from the point of view of optimum signal selection. He also derived some relationships of the Bhattacharyya distance and the divergence function with the Bayesian probability of error. The above mentioned papers formed the basis of future developments in information and distance measures and their use in feature evaluation. An early work to expedite the potentials of the Karhunen-Loeve expansion in feature selection was that of Watanabe [13]. His work has played an important role in the development of mathematical mapping criteria of feature selection. Fu, Min and Li [14] reviewed the contributions made in the first decade of research in the area of feature selection.

Pattern recognition literature published in seventies reflected an increased emphasis on the development of methods of feature selection. This was well deserved. Contributions in the area of feature evaluation included the work of Vilmansen who proposed the use of probabilistic dependence measures in feature evaluation [15]. He also developed some new dependence measures [16]. A fairly exhaustive survey of the contributions to major topics in pattern recognition was made by Kanal [17]. Kittler [18] visualized the mathematical feature selection techniques as belonging to one of two major categories: feature selection in the measurement space and feature selection in the transformed space. Under these two categories he reviewed the roles of probabilistic measures and mathematical mapping methods in feature selection. The survey paper of Chen [19] provided a fairly

complete list of probabilistic measures of feature evaluation and discussed their relationships with the Bayesian error probability. References to specific contributions in the area of probabilistic measures will be made in chapter 2 where these measures are studied individually. Study of mathematical mapping methods is beyond the scope of this thesis. However, in view of the important role played by them in feature selection, here a brief account of the development of the mathematical mapping methods is given and their advantages and disadvantages are compared with that of the probabilistic measures. Because of the simplicity in their implementation the present discussion is confined to linear mapping methods only.

Suppose there is a set of n features from which a subset of r ($1 \leq r \leq n$) features is to be selected. In order to do this using a probabilistic feature evaluation criterion one has to integrate various multivariate probability density functions (involving r features) which themselves are to be estimated from the training data. For $r > 1$ the estimation of a probability density function is a very difficult task. Moreover, one has to repeat the above process of estimation and integration of probability density functions for ${}^{n}C_{r}$ subsets of features. This makes the use of probabilistic measures computationally very demanding.

A mapping method can be described as a process of transformation which, applied on n features, straightway leads to a set of r features. It is worthwhile to note here that, unlike in the case of application of a probabilistic criterion, these r features may not constitute a

subset of the original n features.

An ideal optimality criterion, on the basis of which to derive the transformation procedure, would be the minimization of the Bayesian error probability P_e. Due to difficulty involved in the evaluation of P_e some other heuristic criteria are used which are easily amenable to mathematical treatment. For example, the criteria of minimization of mean square error of representation and maximization of Fisher's discriminant ratio form the basis for the development of a number of wellknown linear mapping methods of feature selection. Mention may be made of Karhunen-Loeve expansion [13], Generalized K-L expansion of Chien and Fu [20], Fukunaga-Koontz transform [21], Discriminant vectors of Foley and Sammon [22]-[24], Kittler-Young criterion [25], [26], Kazakos's criterion [27] and Extended Fisher criterion of Malina [28]. All these measures make use of only the first and the second order moments of the features in different classes.

Because of their computational simplicity the mathematical mapping techniques have been applied in pattern recognition. Though some satisfactory results have been observed in practical applications, the fact remains that, unlike the probabilistic criteria, the mathematical mapping techniques do not have any relationship with the error probability. On the other hand, because of computational difficulties and other estimation problems the probabilistic criteria have remained, except in the case of feature subsets of size 1, more like a subject of theoretical discussion. Under certain simplified assumptions some of the probabilistic criteria have been applied taking into account the feature subsets of size more than one also.

In recent years some efforts have been made to obtain mathematical transformations using the probability of error and other probabilistic measures like the Bhattacharyya coefficient and the divergence function as the optimality criteria [29]-[34]. Though some satisfactory results have been obtained, it is worth noting that in most of these studies Gaussianly distributed features have been considered, in which case the first and the second order moments are sufficient to describe the distributions. In the case of Gaussianly distributed patterns some probabilistic criteria assume closed-form expressions. This makes their computation much simpler because integration is no more involved. These criteria can , therefore, be used for feature evaluation, leading to feature selection, without much difficulty. In the distribution-free case the practical solution to feature selection lies in actually experimenting with various methods to decide which one to choose.

When there is a large number of features to select from, it may be a practical idea to adopt a two-stage process wherein in the first stage a subset of features is selected using a probabilistic criterion evaluating individual features and then, in the second stage, applying a mathmatical mapping criterion on the subset of features selected in the previous stage. The hope is that the use of a mapping criterion in the second stage will give better results than those obtained by the use of a probabilistic criterion evaluating individual features because, in the former method, the interactions between features are also taken into account by way of considering the second order moments.

Because of their relationships with the probability of error the probabilistic criteria have, quite understandably, received a great deal of attention in pattern recognition research. From an analysis of the previous work it is seen that the trend of research has been towards defining new distance functions in order to tighten the existing error bounds or to generalize the existing distance functions. Though some success has been achieved from the point of view of these two objectives, almost all the suggested measures appear to suffer from the same fundamental difficulties that are associated with the direct computation (or estimation) of P_e. In the light of the above difficulties, in the present study more attention has been devoted to search for simple techniques which has led to proposing some Mahalanobis distance-based criteria for feature evaluation.

1.3 Outline of the Thesis

In the present chapter the problem of feature selection is described in the context of pattern recognition and objectives of the thesis are indicated. The chapter continues with a brief account of the early works on probabilistic measures and mathematical mapping methods as used in feature selection. Although the feature selection usually involves both searching and evaluation, searching strategies are considered to be beyond the scope of the present study.

In chapter 2 a critical analysis of the existing probabilistic feature evaluation criteria and their associated error bounds is presented. At the outset it may look like a catalogue of the existing

probabilistic measures. But this cataloging is necessary for an understanding of the trend of research in the development of feature evaluation methodology. Moreover, for a fruitful theoretical comparison of various measures their explicit expressions are needed. This is again true for their experimental investigation. The two-class and the multiclass measures are treated separately. Since, in many cases, the two-class measures form the basis for the development of the multiclass measures, the two-class measures are discussed in greater details.

In chapter 3 the role of the Mahalanobis distance in feature selection is discussed. Following its definition and properties relevant in feature selection two new measures based on the Mahalanobis distance are proposed. In a multiclass pattern recognition problem these new measures are expected to perform better than the direct use of the two-class Mahalanobis distance. Suggestion is also made of the use of some Mahalanobis distance-based statistics in feature selection which have their origin in classical statistical theory of testing of hypothesis.

Chapters 4, 5 and 6 relate to the application of various existing and proposed measures of feature effectiveness in the area of recognition of handprinted numerals.

Chapter 4 deals with the preprocessing steps required for transforming the numerals into a form suitable for implementation of feature evaluation criteria. This includes digitization of data, binarization of character matrices, reduction of noise from the pixels

and extraction of features from the matrices representing the numerals. A preliminary analysis of the extracted features is also made in this chapter which enables discarding the most obvious redundant features.

Chapter 5 deals with feature ordering experiments. To take care of the differences in styles of writing certain numerals these numerals are treated as composed of more than one class, implying that they will require special dealing in the feature ordering and the recognition experiments to follow. To apply the probabilistic measures of feature effectiveness one needs to estimate the class-conditional probability density functions (p.d.f.s) of feature subsets under study. In view of the limited number of character samples in hand, to avoid the estimation problems and to suit the storage requirements, feaure subsets of size one are considered. The class-conditional p.d.f.s for each feature are estimated. All the features are then arranged in decreasing order of their effectiveness, different probabilistic measures leading to different orderings. Feature orderings are also obtained by applying the new Mahalanobis distance-based criteria in two stages. In the first stage the features are arranged by making the simplifying assumption of independence of features and thus computing the values of the Mahalanobis distance only from the means and the standard deviations of the features. A few of the leading features in the above arrangement are then dealt with in the second stage. In this stage the features are rearranged, again by the Mahalanobis distance criteria, but this time taking into account the covariances of the

features. The number of features in the second stage is largely determined by the computational requirements. Rank correlation coefficients between various pairs of feature orderings are computed to get an idea about the conformity of the orderings.

In chapter 6 recognition experiments are described in which different feature orderings obtained in the previous chapter are used. A simple Bayesian classifier, with the assumption of independence of features, is used. Two sets of recognition experiments are considered: in the first all the data are used as both the training data and the test data, and in the second the leave-one-out principle is followed. Analysis of the recognition results leads to a comparative assessment of the methods employed for ordering the features.

In chapter 7 the contrbution made is summarized and suggestions for further research are made. It is emphasized that the Mahalanobis distance, with its computational ease and performance accuracy, deserves more importance in pattern recognition.

Appendices A and B contain the handprinted numeral data and the listings of a number of computer programs, respectively. The programs are written in FORTRAN 77.

CHAPTER 2

PROBABILISTIC MEASURES OF FEATURE EFFECTIVENESS

2.1 Introduction

Pattern classification is a decision-making process in which an input pattern is assigned to one of a number of possible classes depending on the value of the feature vector representing (hopefully!) the input pattern. Suppose there are m ($2 \leq m < \infty$) possible classes $C_1, C_2, \ldots, C_m$ whose a'priori probabilities are $\pi_1, \pi_2, \ldots, \pi_m$, respectively. Suppose the value of the feature vector $X = (X_1, X_2, \ldots, X_n)'$, assumed to be continuous*, taken by the input pattern under consideration is $x = (x_1, x_2, \ldots, x_n)'$. Let Ω_X denote the n-dimensional sample space of X, $p(x|C_i)$ and $P(C_i|x)$ denote the class-conditional probability density function of X given C_i and the a'posteriori probability of class C_i given X=x, respectively, and let $p(x) = \sum_{i=1}^{m} \pi_i . p(x|C_i)$ denote the mixture density of X. Then the Bayesian decision procedure [35], which leads to minimum

---

* If X is discrete then also all the results in the thesis will be valid. The only changes to be made then will be to replace the integration sign by the summation sign in the definitions of various measures and to consider probability mass functions instead of probability density functions.

classification error, assigns an input pattern with feature vector value x to class C_i if $x \in \Omega_i$ where

$$\Omega_i = \{ x \in \Omega_X \mid P(C_i|x) > P(C_j|x), \quad j = 1,2,\ldots,m; j \neq i\}, \quad i = 1,2,\ldots,m \tag{2.1}$$

The corresponding error of misclassification, known as the Bayesian error probability, is given by one of the following expressions:

$$P_e = 1 - E\{\max_i [P(C_i|X)]\} \tag{2.2a}$$

$$= 1 - \int \max_i [P(C_i|x)]p(x)dx \tag{2.2b}$$

$$= 1 - \int \max_i [\pi_i p(x|C_i)]dx \tag{2.2c}$$

where the expectation and the integration are over the sample space Ω_X.

With a feature vector X the minimum achievable error of misclassification is given by P_e. In other words, P_e is an optimum measure of effectiveness of the feature vector X. The question arises as to how (2.2) can be evaluated to yield P_e. There are three obvious alternatives:

(i) the classifier can be built and tested using actual patterns,

or (ii) the integral (2.2) can be evaluated from the probability distributions $P(C_i|x)$, i=1,2, ...,m,

or (iii) if the probability distributions are unknown then some methods such as, for example, curve fitting can be applied to measured data to obtain the forms of $P(C_i|x)$ after which the expression (2.2) is evaluated.

The first alternative involves designing the classifier followed by conducting the classification experiments, thus making the evaluation process slow and costly. In order to avoid this one might think of pursuing either alternative (ii) or alternative (iii). They, however, suffer from the following difficulties:

(i) a closed-form expression of P_e is often not available,

(ii) numerical techniques are often very complex with respect to time and computational needs, and

(iii) partitioning of the feature space Ω_X into Ω_1, Ω_2, ..., Ω_m, which is an indirect prerequisite for performing the integrations of (2.2), is again difficult.

Because of these difficulties various indirect measures have been proposed as a means of approximating P_e and thereby evaluating the effectiveness of a set of features. Most of these indirect measures have been developed based on the concepts of distance, separability, overlap or dependence between the probability distributions characterizing the pattern classes. Some measures have been developed based on information-theoretic considerations. All these measures are expressed in terms of the a'priori probabilities and the probability density functions of the classes. In general they may, therefore, be

termed as probabilistic measures. These measures do not usually bear any exact relationship to P_e and hence upper and lower bounds, expressed in terms of these measures, have been derived to provide an indication of how well they approximate P_e. At this point, it is worth noting that though both the upper and the lower bounds are indicative of how closely a measure approximates P_e, the upper bound is in a sense more useful from the point of view of pattern recognition. If the resulting upper bound is sufficiently low then the pattern recognition system under consideration is 'acceptable'. On the otherhand, lower bound is useful only in the negative sense that if it is sufficiently high then it can lead to a 'rejection' decision. Difference between the upper bound and the lower bound is an indicator of the overall closeness of a measure to P_e.

Many papers dealing with the development of various probabilistic measures and their properties, including relationships with P_e, have appeared during the last two decades. To mention a few, the works of Vajda [36] - [38], Lainiotis [39],[40], Chen [41],[42], Toussaint [43], Vilmansen [15],[16], Chittineni [44]-[48] and Devijver [49],[50] have made a significant contribution to the theoretical development of probabilistic feature evaluation criteria. The following is a list of two-class and multiclass measures that have been suggested as an aid to feature evaluation:

(A) Two-class measures : the Bhattacharyya distance [11],[12],[42]; Jeffreys-Matusita distance function [9],[51]; the Chernoff bound [52],[53]; the Kullback-Leibler numbers [10],[54]; the divergence function [10],[55]; the transformed divergence function [56],[57]; the Kolmogorov variational distance [11]; the generalized

separability measure of Lissack and Fu [58]; the separability measure of Patrick and Fischer [59]; Ito's family of approximating functions [60]; Toussaint's measure of affinity [61],[62]; the f-divergence of Csiszar [63] and the χ^{α}-divergence of Vajda [64].

(B) Multiclass measures : Matusita's measure of affinity [65]-[69]; Devijver's generalized divergence measure [70]; Toussaint's generalized Kolmogorov variational distance [71]; Shannon's conditional entropy (or Equivocation) [53],[72],[73]; Mutual information [5]-[7],[42],[74]-[79]; the Bayesian distance of Devijver [49]; the conditional quadratic entropy of Vajda [36],[80]; the conditional cubic entropy of Chen [19]; Renyi's conditional entropy of order α [81],[82]; Minkowski's measure of nonuniformity [43],[49]; probabilistic dependence measures of Vilmansen [15],[16],[18]; and generalized distance measures of Backer, etal. [83],[84].

The above list is by no means exhaustive . Any well-behaved function of the probability density functions of the classes which provides some measure of separability between the classes may be considered for feature evaluation. In the following four sections of this chapter a theoretical study of the probabilistic measures is presented, concentrating only on those measures that are widely used and/or appear to be of most value.

2.2 Two-class Measures and Their Associated Error Bounds

2.2.1 The Bhattacharyya Distance

The Bhattacharyya distance, b, proposed by Kailath [11], is defined in terms of the Bhattacharyya coefficient [12], ρ, as follows:

$$b = - \ln \rho \tag{2.3}$$

where

$$\rho = \int [p(x|C_1).p(x|C_2)]^{\frac{1}{2}} dx \tag{2.4}$$

Clearly, $0 \leq \rho \leq 1$ and so $0 \leq b \leq \infty$. On taking the a'priori probabilities into account, the Bhattacharyya coefficient can be generalized so that it becomes [55],[60]

$$\rho^* = \int [\pi_1 p(x|C_1).\pi_2 p(x|C_2)]^{1/2} dx \tag{2.5a}$$

$$= E \left\{ [P(C_1|X).P(C_2|X)]^{1/2} \right\} \tag{2.5b}$$

It is easy to see that ρ^* and ρ are related by the following exact relationship:

$$\rho^* = \sqrt{\pi_1 \pi_2} \cdot \rho \tag{2.6}$$

The results given in terms of ρ can, therefore, also be expressed in terms of ρ^*.

Hudimoto [85],[86] showed that P_e is bounded above and below by the following relationships:

$$\pi_1\pi_2\rho^2 \leq \frac{1}{2} - \frac{1}{2}\sqrt{1 - 4\pi_1\pi_2\rho^2} \leq P_e \leq \sqrt{\pi_1\pi_2}\ \rho \tag{2.7}$$

The difference between the upper and the tighter lower bound is given by

$$\delta = \sqrt{\pi_1\pi_2}\ \rho - \frac{1}{2} + \frac{1}{2}\sqrt{1 - 4\pi_1\pi_2\rho^2} \tag{2.8}$$

In the following theorem an interesting property of δ is proved for the first time.

Theorem 2.1 Whatever be the values of the a'priori probabilities π_1 and π_2, (i) the maximum value of δ is $\frac{1}{2}(\sqrt{2}-1)$ and (ii) this maximum cannot be attained for π_1 values (and so π_2 values) lying outside the interval $\left(\frac{2-\sqrt{2}}{4}, \frac{2+\sqrt{2}}{4}\right)$.

Proof (i) Taking the first derivative of δ with respect to ρ one gets,

$$\frac{d\delta}{d\rho} = \sqrt{\pi_1\pi_2} - \frac{2\pi_1\pi_2\rho}{\sqrt{1 - 4\pi_1\pi_2\rho^2}}$$

Equating the above expression to 0 leads to

$$\rho = \frac{1}{\sqrt{8\pi_1\pi_2}} \tag{2.9}$$

Thus the maximum value of δ is attained at the above value of ρ. Substituting this value of ρ in equation (2.8) gives,

$$\begin{aligned}\delta_{max} &= \sqrt{\pi_1\pi_2}\cdot\frac{1}{\sqrt{8\pi_1\pi_2}} - \frac{1}{2} + \frac{1}{2}\sqrt{1 - 4\pi_1\pi_2\cdot\frac{1}{8\pi_1\pi_2}} \\ &= \frac{1}{\sqrt{8}} - \frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{1}{2}} \\ &= \frac{1}{2}(\sqrt{2} - 1).\end{aligned}$$

Hence the first part of the theorem is proved.

(ii) The above mentioned maximum value occurs at $\rho = \frac{1}{\sqrt{8\pi_1\pi_2}}$. But ρ is restricted by the condition $\rho \leq 1$. Solution of the inequality $\rho \leq 1$ leads to the desired result. Q.E.D.

In Figures 2.1a to 2.1d the P_e bounds for different values of π_1, namely, $\pi_1 = 0.500, 0.625, 0.750$ and 0.875, are shown. It can be seen from these figures that the value of ρ for which the maximum value of δ ($=\frac{1}{2}(\sqrt{2}-1) \doteq 0.2071$) is attained gets shifted towards the right with increases in the value of π_1. For values of $\pi_1 > \frac{2+\sqrt{2}}{4}$ the maximum occurs at a value of ρ outside its range (Fig. 2.1d).

Use of an indirect measure as an alternative to P_e is justified provided (i) its bounds are sufficiently close to P_e and (ii) it is easier to compute than P_e. As shown above, the maximum difference between the two bounds can be as large as 0.2071. Therefore, from the

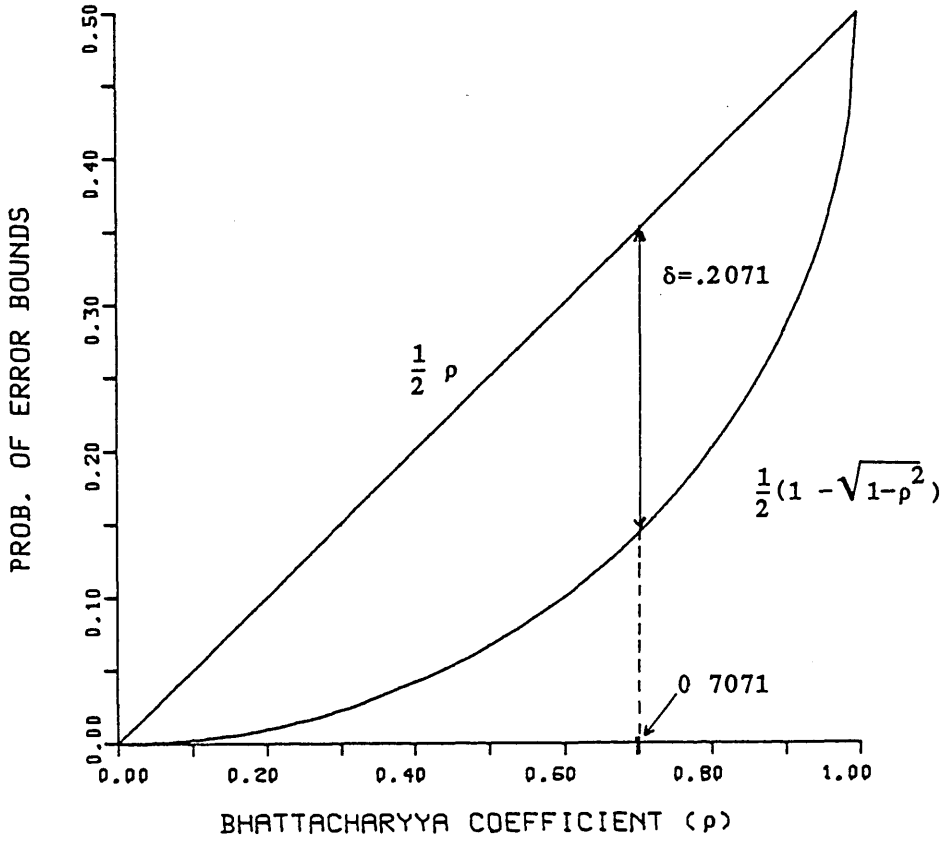


Fig. 2.1a Prob. of error (P_e) bounds in terms of the Bhattacharyya coefficient (ρ) for $\pi_1=\pi_2=0.500$

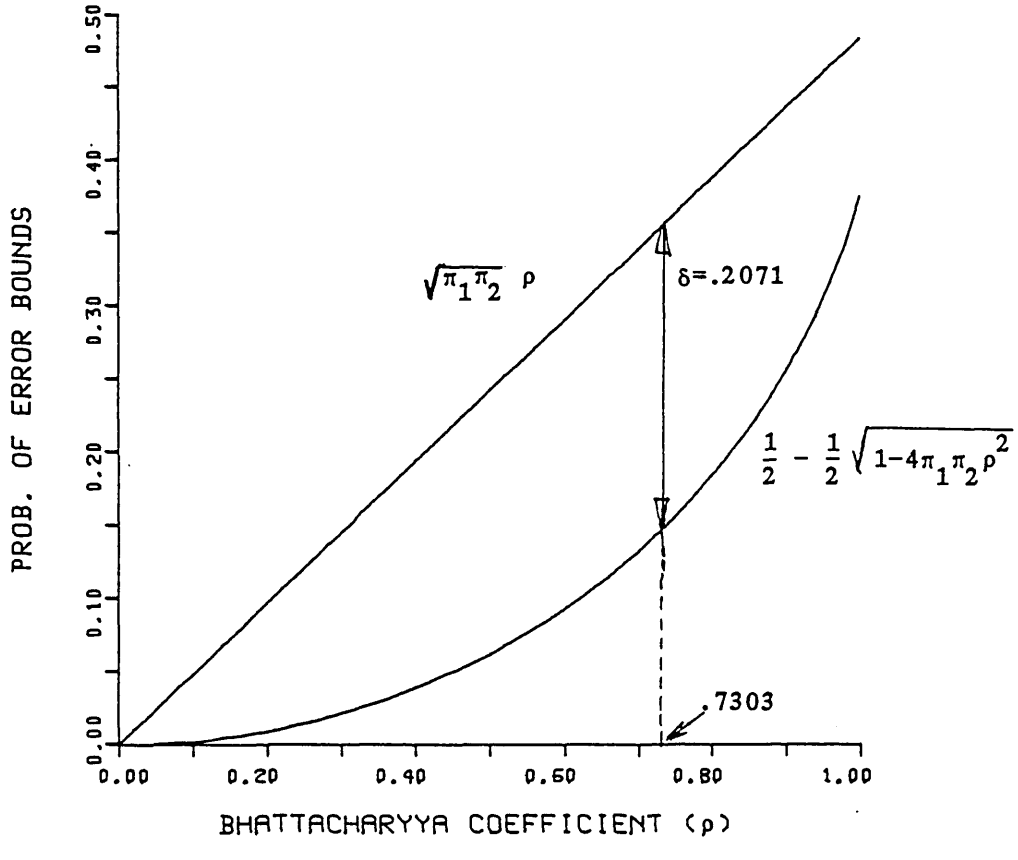


Fig. 2.1b Prob. of error (P_e) bounds in terms of the Bhattacharyya coefficient (ρ) for $\pi_1=0.625$, $\pi_2=0.375$

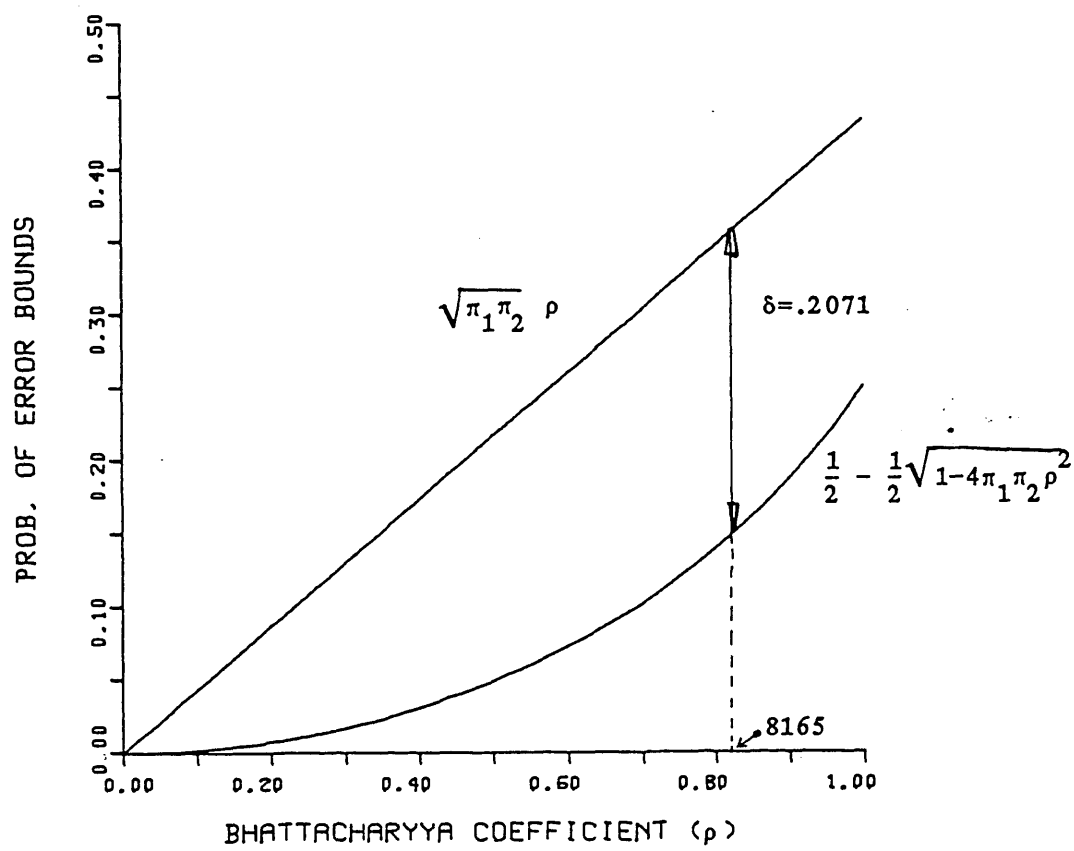


Fig. 2.1c Prob. of error (P_e) bounds in terms of the Bhattacharyya coefficient (ρ) for π_1=0.750 and π_2=0.250

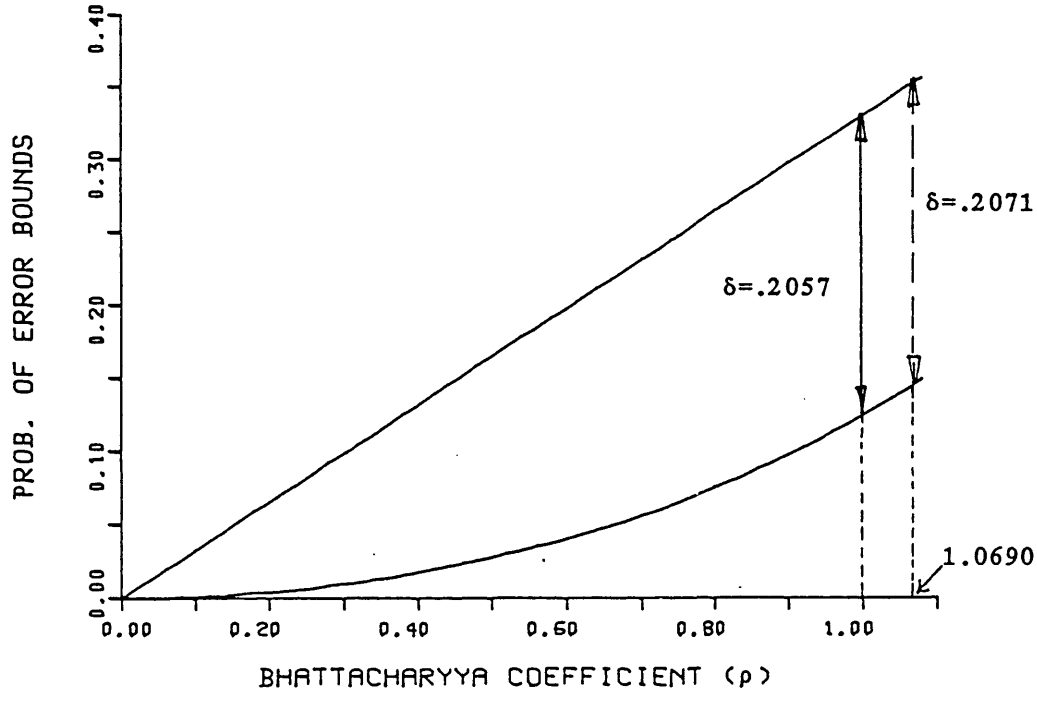


Fig. 2.1d Prob. of error (P_e) bounds in terms of the Bhattacharyya coefficient (ρ) for π_1=0.875 and π_2=0.125

point of view of pattern recognition, one may not like to consider the Bhattacharyya distance b to be a close approximation to P_e. But b has a computational advantage over P_e. For the exponential family of distributions b assumes a closed-form expression. In this case, therefore, numerical integration is not involved in the computation of b. Explicit expressions of b are available [11] for the general exponential family and also for a few special cases of this family, namely, the Multinomial distribution, the Poisson distribution, and the Gaussian distribution. For the Gaussian distributions described by $p(x|C_i) = N(\mu_i, V_i)$, i=1,2 b assumes the following expression:

$$b = \frac{1}{8}(\mu_1 - \mu_2)' \, V^{-1} \, (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{\det (V)}{\sqrt{\det (V_1) \, . \, \det (V_2)}}$$

where

$$V = (V_1 + V_2) \, / \, 2 \qquad (2.10)$$

In most cases, however, a closed-form expression does not exist and therefore the computation of b becomes as difficult as that of P_e.

2.2.2 The Jeffreys-Matusita Distance Function [9],[51]

The Jeffreys-Matusita distance function, γ, is defined to be

$$\gamma = \left[\int \{ \sqrt{p(x|C_1)} - \sqrt{p(x|C_2)} \}^2 dx \right]^{1/2} \tag{2.11}$$

and when the a'priori probabilities are taken into account

$$\gamma^* = \left[\int \{ \sqrt{\pi_1 p(x|C_1)} - \sqrt{\pi_2 p(x|C_2)} \}^2 dx \right]^{1/2} \tag{2.12a}$$

$$= \left[E \{ \sqrt{P(C_1|X)} - \sqrt{P(C_2|X)} \}^2 \right]^{1/2} \tag{2.12b}$$

Though, apparently, the Jeffreys-Matusita distance was defined without the knowledge of the existence of the Bhattacharyya coefficient, they bear the following exact relationships:

$$\gamma = \sqrt{2 \ (1 - \rho)} \tag{2.13a}$$

and

$$\gamma^* = \sqrt{1 - 2\rho^*} \tag{2.13b}$$

From these exact functional relationships it can be seen that they are two different versions of the same measure and, hence, there is no need to consider separately the properties of the

Jeffreys-Matusita distance function. It is only necessary to note that this distance function has the same advantages and disadvantages as are associated with the Bhattacharyya coefficient.

2.2.3 The Divergence Function

The divergence function was first introduced by Jeffreys [9],[87]. It is defined by

$$J = \int [p(x|C_1) - p(x|C_2)] \ln \left[\frac{p(x|C_1)}{p(x|C_2)} \right] dx \qquad (2.14)$$

and can be looked upon as a symmetrical form of the two Kullback-Leibler numbers [88] obtained from the addition of the numbers. The following modified version of the divergence function, incorporating the a'priori probabilities, has been provided by Toussaint [55]:

$$J^* = \int \left[\pi_1 p(x|C_1) - \pi_2 p(x|C_2) \right] \ln \left[\frac{\pi_1 p(x|C_1)}{\pi_2 p(x|C_2)} \right] dx \qquad (2.15a)$$

$$= E \left\{ [P(C_1|X) - P(C_2|X)] \ln \left[\frac{P(C_1|X)}{P(C_2|X)} \right] \right\} \qquad (2.15b)$$

It is easy to see that both J and J^* can have values in the interval $[0,\infty)$ and that they are related by $J^*(1/2,1/2) = J/2$. A

number of lower bounds to P_e, expressed in terms of J and J^*, have been discovered by Kailath [11] and Toussaint [55]. Among those bounds which can be expressed in terms of J^* the following bound of Toussaint [40] is the tightest:

$$P_e \geq \frac{1}{2} - \frac{1}{2}\sqrt{1 - 4 \exp\ [-2H(\pi) - J^*]} \tag{2.16}$$

where

$$H(\pi) = -\pi_1 \ln \pi_1 - \pi_2 \ln \pi_2 \tag{2.17}$$

Fig. 2.2 gives the diagrammatic representation of (2.16) for different values of π_1 (and π_2).

For $\pi_1 = \pi_2 = 1/2$ (2.16) reduces to

$$P_e \geq \frac{1}{2} - \frac{1}{2}\sqrt{1 - \exp(-J/2)} \tag{2.18}$$

Toussaint [55] also derived the following inequality between J^* and P_e:

$$J^* \geq (2 P_e - 1). \ln \left[\frac{P_e}{1 - P_e}\right] \tag{2.19}$$

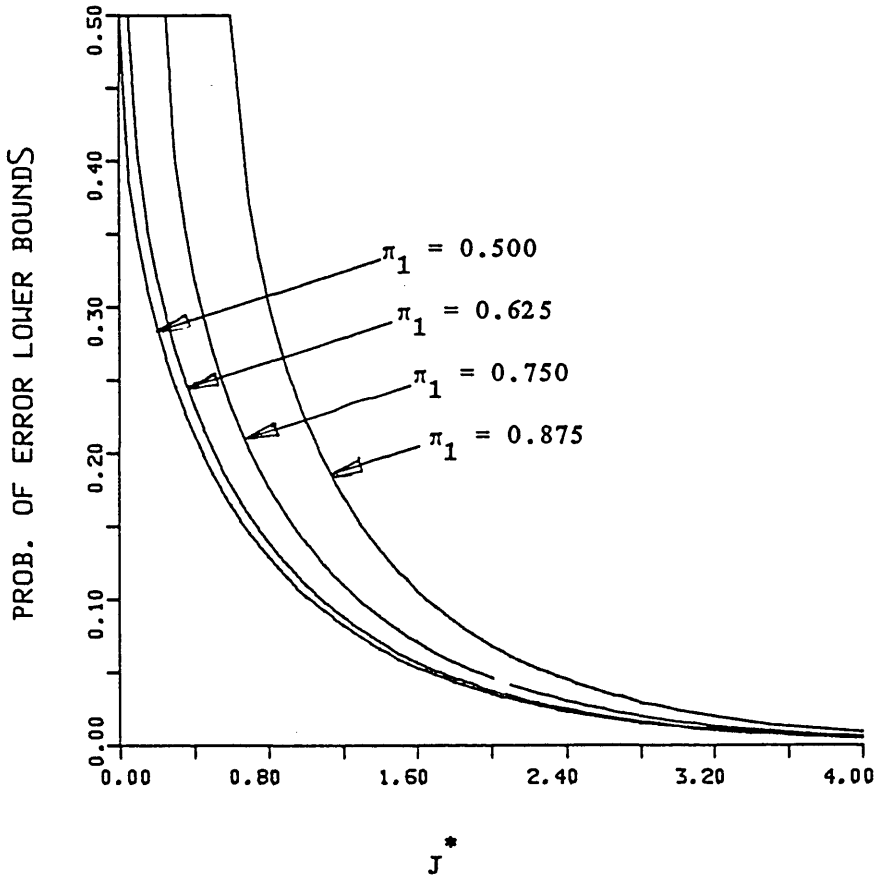


Fig. 2.2 Prob. of error (P_e) lower bounds versus J^* for different values of π_1

For equal a'priori prababilities (2.19) reduces to

$$J \geq 2\ (2\ P_e - 1).\ \ln\ \left[\frac{P_e}{1 - P_e}\right] \tag{2.20}$$

(2.20) provides the tightest inequality between J and P_e but has the drawback that P_e cannot be expressed in terms of J. Fig. 2.3 illustrates the improved tightness of (2.20) over (2.18).

It is understood that there cannot be a general distribution-free upper bound of P_e which is expressable in terms of the divergence function [89]. For the Gaussian distribution an upper bound is available [90] and this is given by

$$P_e \leq \sqrt{\pi_1 \pi_2}\ \left(\frac{J}{4}\right)^{-1/4} \leq \frac{1}{2}\ \left(\frac{J}{4}\right)^{-1/4} \tag{2.21}$$

As for b, J can also be expressed in closed-form expression for the exponential family of distributions. For $p(x|C_i) = N(\mu_i, V_i), i=1,2$ it reduces to

$$J = \frac{1}{2}\ \mathrm{tr}\ \ [V_1^{-1}V_2 + V_2^{-1}V_1 - 2\ I]$$

$$+ \frac{1}{2}\ \mathrm{tr}[(V_1^{-1} + V_2^{-1})\ (\mu_1 - \mu_2)\ (\mu_1 - \mu_2)'] \tag{2.22}$$

In those cases for which a closed-form expression can be found for J, it provides an effective feature evaluation criterion.

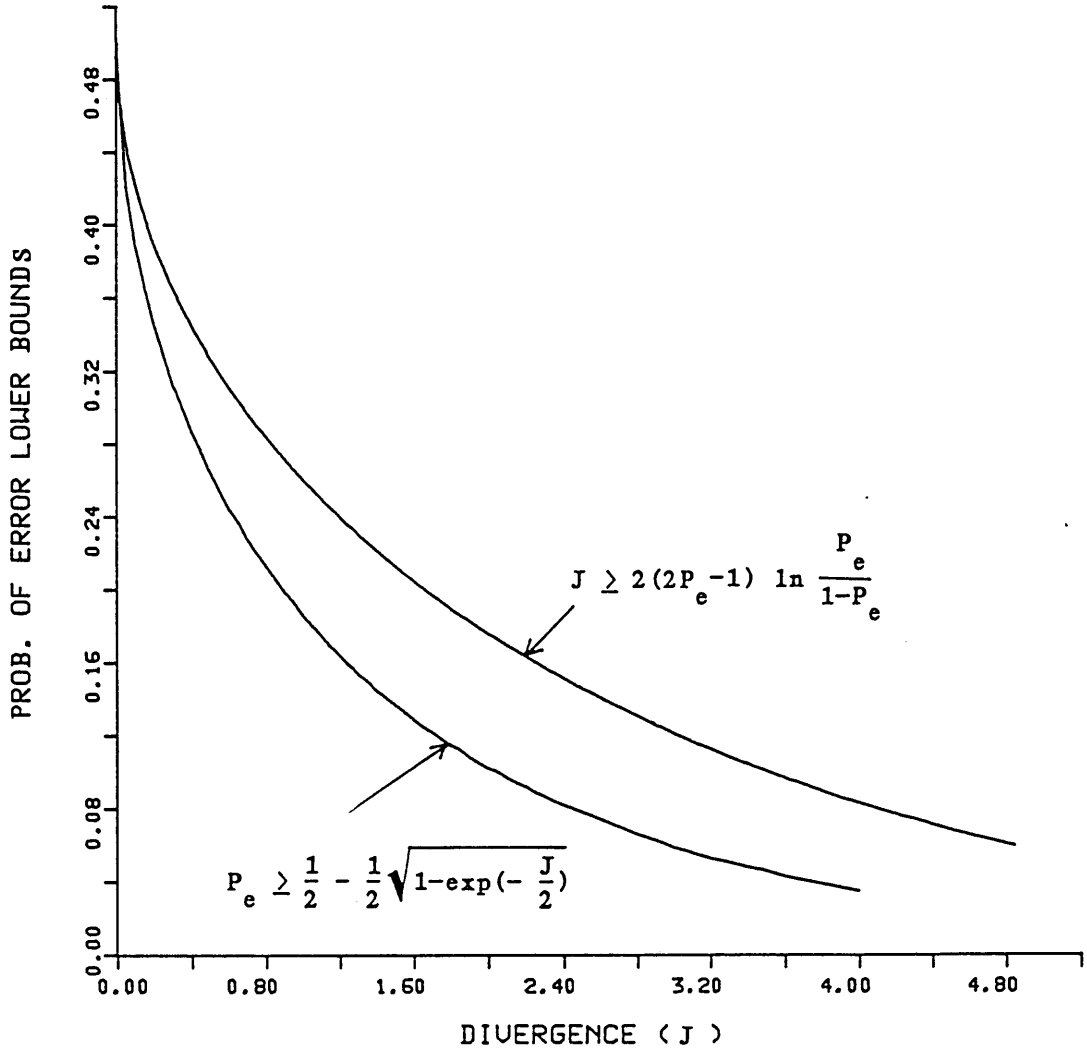


Fig. 2.3 Prob. of error (P_e) lower bounds versus J

In the case of multivariate Gaussian distribution the computational complexities of the Bhattacharyya distance and the divergence function can be compared from the expressions (2.10) and (2.22). Computationally, the determination of V^{-1} is the same as the determination of det(V). Apart from matrix inversion and determinant calculation, all other arithmetic operations present in (2.10) and (2.22) are computationally less complex. The number of computationlly complex operations present in (2.10) is three, namely, $\det(V_1)$, $\det(V_2)$ and det(V) whereas the computation of J involves only two, namely, V_1^{-1} and V_2^{-1}. The conclusion can thus be drawn that in the case of the Gaussian distribution the divergence function is computationally more effective than the Bhattacharyya distance. However, the upper bound of P_e in terms of ρ and the upper bound in terms of J are related as follows:

$$P_e \leq \sqrt{\pi_1\pi_2}\,\rho \leq \sqrt{\pi_1\pi_2}\,\left(\frac{J}{4}\right)^{-1/4} \tag{2.23}$$

It can thus be seen that J provides a less tight upper bound than ρ, which offsets its computational advantage.

2.2.4 Transformed Divergence

The transformed divergence, J_T, defined by Swain etal.[56],[57], is given by

$$J_T = 2\ [\ 1 - \exp(-\frac{J}{8})\ ] \tag{2.24}$$

Because of the exact relationship between J and J_T, the error

bounds expressed in terms of J can also be expressed in terms of J_T. In Fig. 2.4 the relationship between J_T and the lower bounds of P_e is shown. This diagram is plotted using the equation (2.24) and the following inequality:

$$P_e \geq \frac{1}{2} - \frac{1}{2}\sqrt{1 - \exp(-\frac{J}{2})} \qquad (2.25)$$

which is obtained by putting $\pi_1=\pi_2=1/2$ in (2.16).

For increasing separability between classes the divergence J increases in an unbounded fashion whereas J_T increases in a bounded manner, with its maximum value being equal to 2. Although multiclass systems are to be considered later, it is appropriate to remark at this point that in a multiclass problem, solved through the use of an average of the pairwise values of J_T, this bounded characteristics helps to prevent highly separable class-pairs from making an undue contribution to the average separability criterion. Therefore, in a multiclass situation the use of J_T, rather than J, is recommended. In a two-class case, however, the unbounded nature of J with increasing separability does not affect the feature ordering.

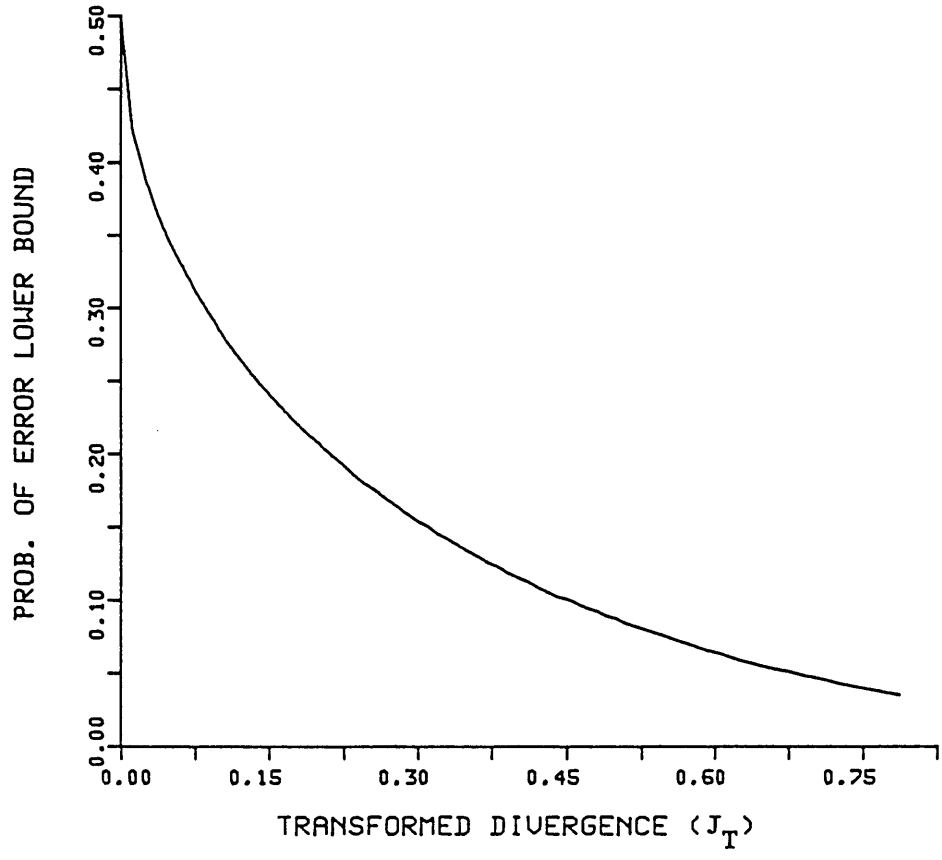


Fig. 2.4 Prob. of error (P_e) lower bound versus J_T

2.2.5 Kolmogorov Variational Distance

The Kolmogorov variational distance [11], K, is defined to be

$$K = \frac{1}{2}\int |\pi_1 p(x|C_1) - \pi_2 p(x|C_2)|\,dx \tag{2.26a}$$

$$= \frac{1}{2}E\{|P(C_1|X) - P(C_2|X)|\} \tag{2.26b}$$

It can be shown [91] that

$$P_e = \frac{1}{2} - K \tag{2.27}$$

Thus from the variational distance of Kolmogorov the Bayesian probability of error can be determined exactly. But the difficulty associated with the use of K is that its computational complexity is precisely the same as that of P_e.

2.2.6 Generalized Separability Measure of Lissack and Fu

The generalized separability measure proposed by Lissack and Fu [58] is defined as

$$K_\alpha = E\{|P(C_1|x) - P(C_2|x)|^\alpha\}, \quad 0 < \alpha < \infty. \tag{2.28}$$

This is a straightforward generalization of the Kolmogorov variational distance, and for $\alpha = 1$ the measure reduces to 2K. In this case, therefore,

$$K_1 = 2K = 1 - 2P_e \tag{2.29}$$

Error bounds for different values of α are as follows:

For $0 < \alpha \leq 1$,

$$\frac{1}{2}\{1 - K_\alpha\} \leq P_e \leq \frac{1}{2}\{1 - [K_\alpha]^{1/\alpha}\} \tag{2.30}$$

and for $1 \leq \alpha < \infty$,

$$\frac{1}{2}\{1 - [K_\alpha]^{1/\alpha}\} \leq P_e \leq \frac{1}{2}\{1 - K_\alpha\} \tag{2.31}$$

For $\alpha = 1$ the lower and the upper bounds coincide and an increase or decrease in its value loosens the bounds. From the following theorem, proved for the first time, one can obtain information about the magnitude of the loosening of the bounds for $\alpha > 1$.

Theorem 2.2 (i) For a given $\alpha > 1$ the maximum value of δ (= upper bound - lower bound) is given by

$$\delta_{max} = \frac{1}{2} \{ \alpha^{-\frac{1}{\alpha-1}} - \alpha^{-\frac{\alpha}{\alpha-1}} \}$$

and (ii) the value of δ_{max} increases with increase in α.

Proof (i) For $\alpha > 1$,

$$\delta = \frac{1}{2}\{1-K_\alpha\} - \frac{1}{2}\{1-[K_\alpha]^{1/\alpha}\}$$

$$= \frac{1}{2}\{[K_\alpha]^{1/\alpha} - K_\alpha\} \qquad (2.32)$$

Differentiating δ with respect to K_α and equating the derivative to zero gives

$$K_\alpha = \alpha^{-\frac{\alpha}{\alpha-1}} \qquad (2.33)$$

Thus the maximum value of δ occurs for the above value of K_α. Putting this value of K_α in (2.32) gives

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

$$= \frac{1}{2} \left\{ \alpha^{-\frac{1}{\alpha-1}} - \alpha^{-\frac{\alpha}{\alpha-1}} \right\}$$

(ii) Differentiating δ_{max} with respect to α,

$$\frac{d\delta_{max}}{d\alpha} = \frac{1}{2}\left\{ \alpha^{-\frac{1}{\alpha-1}} \left[\frac{\log \alpha}{(\alpha-1)^2} - \frac{1}{(\alpha-1)\alpha}\right] - \alpha^{-\frac{\alpha}{\alpha-1}} \left[\frac{\log \alpha}{(\alpha-1)^2} - \frac{1}{\alpha-1}\right] \right\}$$

$$= \frac{1}{2}\left\{ \frac{\log \alpha}{(\alpha-1)^2} \left[\alpha^{-\frac{1}{\alpha-1}} - \alpha^{-\frac{\alpha}{\alpha-1}}\right] + \frac{1}{\alpha-1}\left[\alpha^{-\frac{\alpha}{\alpha-1}} - \frac{1}{\alpha}\alpha^{-\frac{1}{\alpha-1}}\right] \right\} \tag{2.34}$$

Using the identity $\frac{\alpha}{\alpha-1} = 1 + \frac{1}{\alpha-1}$ in (2.34) leads to

$$\frac{d\delta_{max}}{d\alpha} = \frac{1}{2}\left\{ \frac{\log \alpha}{(\alpha-1)^2} \alpha^{-\frac{1}{\alpha-1}} (1 - \frac{1}{\alpha}) + \frac{1}{\alpha-1}\left[\frac{1}{\alpha}\alpha^{-\frac{1}{\alpha-1}} - \frac{1}{\alpha}\alpha^{-\frac{1}{\alpha-1}}\right] \right\}$$

$$= \frac{1}{2}\left\{ \frac{\log \alpha}{(\alpha-1)^2} \alpha^{-\frac{1}{\alpha-1}} (1 - \frac{1}{\alpha}) \right\} \tag{2.35}$$

It is easy to see that the expression in the right hand side of (2.35) is +ve. Hence the desired result is proved. Q.E.D.

For a given $\alpha > 1$ the upper and the lower bounds of P_e corresponding to the maximum difference between the two bounds are given by

$$P_e^U = \frac{1}{2}\{ 1 - \alpha^{-\frac{\alpha}{\alpha-1}} \} \tag{2.36}$$

and

$$P_e^L = \frac{1}{2}\{ 1 - \alpha^{-\frac{1}{\alpha-1}} \} \tag{2.37}$$

Fig. 2.5 shows how the values of P_e^U and P_e^L vary with α. It may be noted that as α increases from 1 to ∞ the maximum difference between the two bounds increases from 0 to 0.5. This shows how the

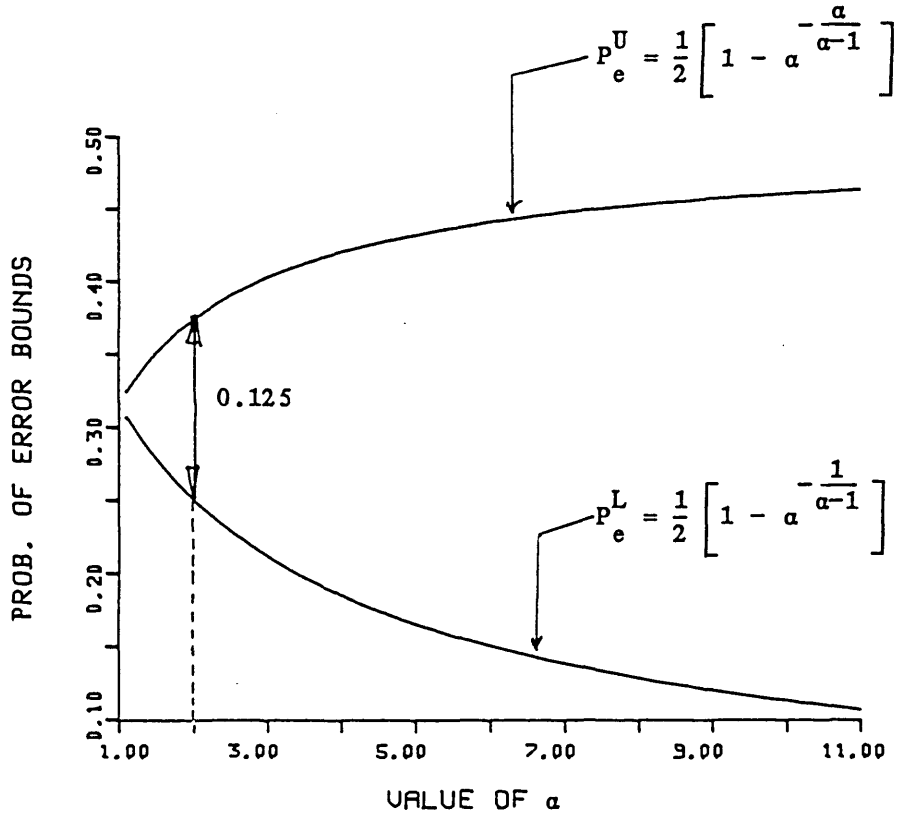


Fig. 2.5 Looseness in prob. of error (P_e) bounds given by K_α for different values of $\alpha \geq 1$

bounds loosen with increasing α. Moreover, with increasing α the computation of K_α becomes more demanding. Therefore, it appears that there is no advantage in going for high values of α. The bounds corresponding to K_2 (α=2) are tighter than most of the existing bounds associated with the other two-class measures. As can be seen from Fig. 2.5, the maximum difference between the two bounds in this case is 0.125. K_2 has the advantage over K_1 in that K_2 involves the operation of raising $P(C_1|X) - P(C_2|X)$ to the power of 2 which is mathematically more handy to deal with than the difference operation involved in K_1. But as is the case with K_1; K_2 also cannot be expressed in a closed-form even for the Gaussian distribution. This renders K_2 computationally ineffective.

2.2.7 Ito's Family of Approximating Functions

The approximating functions of Ito [60] are defined as

$$Q_r = \frac{1}{2} - \frac{1}{2}\left[E\left\{ [\, P(C_1|X) - P(C_2|X)\,]^{\frac{2(r+1)}{2r+1}} \right\} \right], \qquad r = 0, 1, 2, \ldots \tag{2.38}$$

Ito has shown that P_e is upper bounded by the following relationship:

$$P_e \leq Q_r \leq \ldots \leq Q_0 \tag{2.39}$$

Though Q_r provides tighter upper bounds for larger values of r, it cannot be expressed in closed-form even for Gaussian distribution and hence it suffers from the same computational difficulties as mentioned previously.

Except for $r = 0$ the Q_r function does not provide a lower bound to P_e. It is easy to see that

$$Q_0 = \frac{1}{2}[1 - K_2] \tag{2.40}$$

Using this relationship and putting $\alpha = 2$ in (2.31) gives the following P_e bounds in terms of Q_0:

$$\frac{1}{2}[1 - \sqrt{1 - 2Q_0}] \leq P_e \leq Q_0 \tag{2.41}$$

It may be noted that the upper bound in (2.41) coincides with that in (2.39). In Fig. 2.6 the bounds mentioned in (2.41) are depicted. The maximum difference of 0.125 between the two bounds occurs at $Q_0 = 0.375$.

Ito proved that

$$Q_0 = E_1 \tag{2.42}$$

where E_1 is the error probability of the first Nearest Neighbor (1NN) decision rule of Cover and Hart [92]. This result is interesting because, in the nonparametric case, Q_0 can be estimated by classifying the sample data with 1NN rule, thus making Q_0 a useful nonparametric feature evaluation criterion.

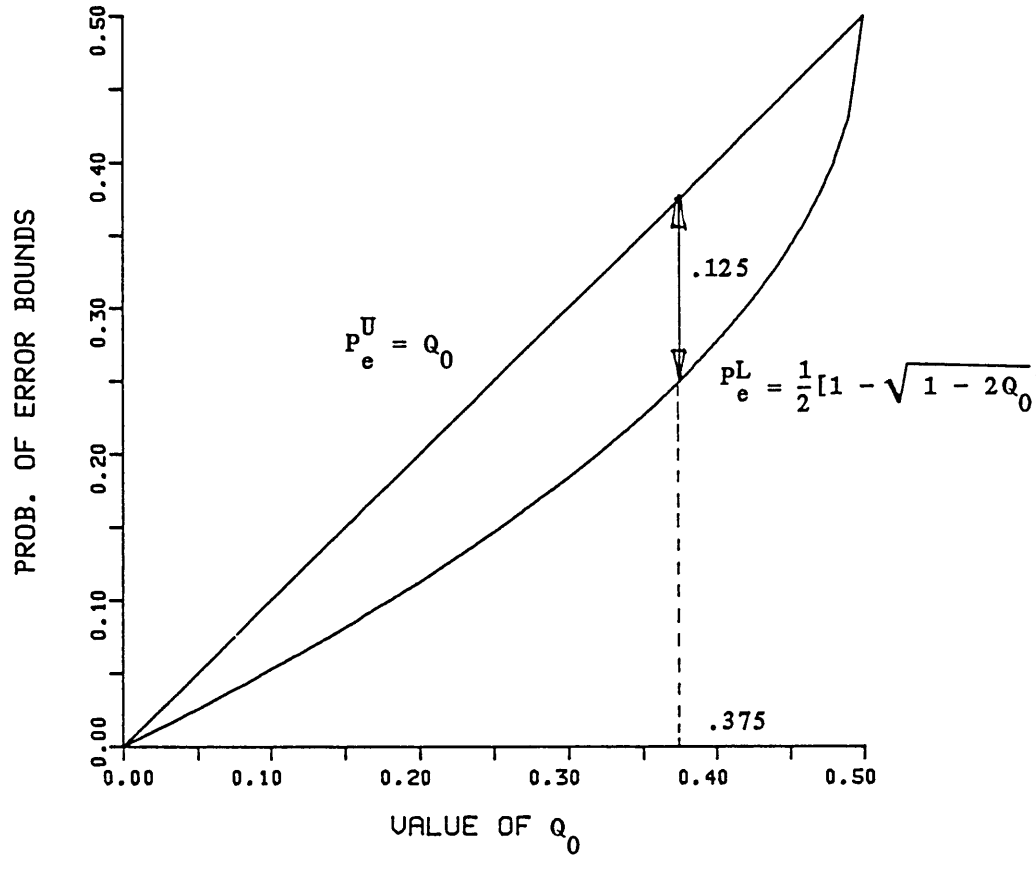


Fig. 2.6 Prob. of error (P_e) bounds versus Ito's Q_0-function

2.2.8 Toussaint's Measures of Affinity

Toussaint's affinity measure [62], which is analogous to the Bhattacharyya coefficient, is defined by

$$\tau_r = 2^{\frac{1}{r}} \int \frac{\pi_1 p(x|C_1) \cdot \pi_2 p(x|C_2)}{\left\{ [\pi_1 p(x|C_1)]^r + [\pi_2 p(x|C_2)]^r \right\}^{1/r}} dx,$$

$$r = 1,2, \ldots \qquad (2.43)$$

The associated distance function is

$$T_r = - \ln (2\tau_r) \qquad (2.44)$$

Toussaint [62] showed that

$$P_e = \tau_\infty \leq \ldots \leq \tau_r \leq \ldots \leq \tau_1 \qquad (2.45)$$

Thus by increasing the value of r tighter upper bounds can be obtained. The difficulty of computation remains, however, since τ_r does not simplify for parametric cases.

The following result of Toussaint shows that τ_1 is more closely related to P_e than is the Bhattacharyya coefficient ρ:

$$P_e \leq \tau_1 \leq \sqrt{\pi_1 \pi_2}\,\rho \tag{2.46}$$

As is the case with Q_0, τ_1 is also equal to 1NN error rate, thus making it useful in a nonparametric situation.

2.3 Multiclass Measures and Their Associated Error Bounds

2.3.1 Matusita's Measure of Affinity

Matusita's measure of affinity [66] is an m-class generalization of the Bhattacharyya coefficient and is defined by

$$\rho_m = \int [\, p(x|C_1)\, p(x|C_2) \ldots p(x|C_m)\,]^{1/m}\, dx \tag{2.47}$$

Taking into account the a'priori probabilities the measure of affinity can be generalized [68] as follows:

$$\rho_m^* = \int [\, \pi_1 p(x|C_1)\, \pi_2 p(x|C_2) \ldots \pi_m p(x|C_m)\,]^{1/m}\, dx \tag{2.48a}$$

$$= E\left\{ [\, P(C_1|X)\, P(C_2|X) \ldots P(C_m|X)\,]^{1/m} \right\} \tag{2.48b}$$

Clearly,

$$\rho_m^* = [\pi_1\pi_2 \ \ldots \ \pi_m]^{1/m} \ \rho_m \tag{2.49}$$

Since ρ_m^* is explicitly expressable in terms of ρ_m, error bounds can also be given in terms of ρ_m.

Toussaint [69] obtained the following upper bound of P_e:

$$P_e \leq \frac{m}{2} \left[1 + (\pi_1\pi_2 \ldots \pi_m)^{1/m} \ \rho_m\right] - 1 \tag{2.50}$$

This appears to be the only upper bound, in terms of ρ_m, available in the literature. For m = 2 it reduces to the wellknown Hudimoto upper bound given in (2.7).

Toussaint [68] has shown that P_e can be lower bounded as follows:

$$P_e \geq \frac{1}{2} - \frac{1}{2}\sqrt{1 - 4(m-1)^{m-1} \ \pi_1\pi_2 \ldots \pi_m \ \rho_m^m} \tag{2.51}$$

The form of this bound is similar to the tighter lower bound in (2.7) and for m = 2 it reduces to that bound.

Since for m = 2 the bounds given by (2.50) and (2.51) coincide with the bounds mentioned in (2.7), for their graphical representation reference may be made to Figures 2.1a to 2.1d shown earlier. For higher values of m the upper bound (2.50) becomes loose very rapidly and for $m \geq 4$ it becomes useless because then its value exceeds 1.0.

If the a'priori probabilities are equal then from (2.50) and (2.51) one gets

$$\frac{1}{2} - \frac{1}{2}\sqrt{1 - \frac{4(m-1)^{m-1}}{m^m}\rho_m^m} \leq P_e \leq \frac{m}{2}\left[1 + \frac{\rho_m}{m}\right] - 1 \qquad (2.52)$$

In Fig. 2.7 the above bounds are plotted for m = 3. It is easy to see that in an m-class situation with equal a'priori probabilities the error probability is bounded above as follows:

$$P_e \leq 1 - \frac{1}{m} \qquad (2.53)$$

Comparing the upper bound mentioned in (2.52) with the trivial upper bound (2.53) one can see that even for m = 3 the upper bound in (2.52) becomes useless when ρ_m (m = 3) exceeds the value of 1/3. The trivial upper bound is shown in dotted line in Fig. 2.7. As can be seen in this diagram, the gap between the upper and the lower bound is quite large.

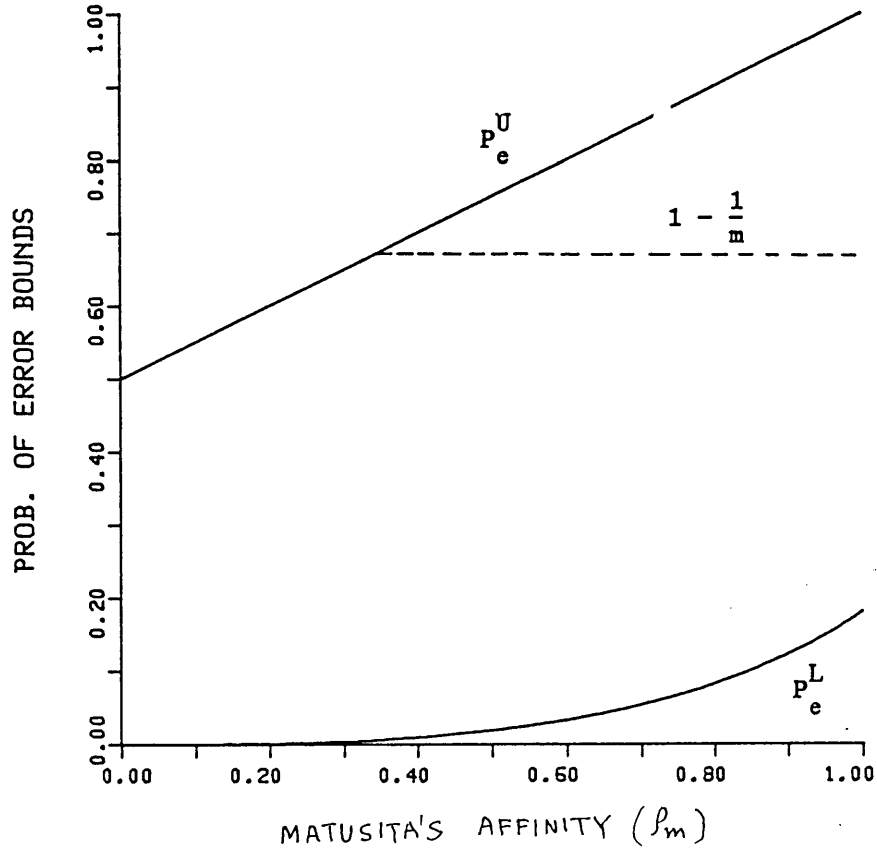


Fig. 2.7 Prob. of error (P_e) bounds versus Matusita's m-class affinity (ρ_m^e) for m=3 and $\pi_1=\pi_2=\pi_3=1/3$

2.3.2 Shannon's Conditional Entropy

With C denoting the set of classes $C_1, C_2, \ldots, C_m$ the equivocation or Shannon's conditional entropy [72],[73] of C given X is defined by*

$$H = H(C|X) = \int \left[-\sum_{i=1}^{m} P(C_i|x) \ln P(C_i|x) \right] p(x)\, dx \qquad (2.54a)$$

$$= E\left\{ -\sum_{i=1}^{m} P(C_i|X) \ln P(C_i|X) \right\} \qquad (2.54b)$$

The wellknown [93] upper bound on P_e in terms of H is

$$P_e \leq \frac{1}{2 \ln 2} H, \quad m \geq 2 \qquad (2.55)$$

For m = 2 it can be shown [41] that

$$P_e \leq \frac{1}{2 \ln 2} H \leq \sqrt{\pi_1 \pi_2}\ \rho \qquad (2.56)$$

Thus H gives tighter upper bound than ρ. But H is computationally less efficient than ρ.

---

* In this thesis natural logarithm is used instead of logarithm to the base 2 considered by Shannon.

A piecewise linear upper bound, obtained apparently independently by Kovalevsky [73] and Tebbe and Dwyer [94], gives a tighter bound than that given by (2.55) above. This piecewise linear bound is given by

$$H \geq \ln(r) + r(r+1)\left(\ln\frac{r+1}{r}\right)\left(P_e - \frac{r-1}{r}\right),\ m \geq 2 \qquad (2.57)$$

where r is such an integer that

$$\frac{r-1}{r} \leq P_e < \frac{r}{r+1} \qquad (2.58)$$

For $P_e \leq 0.5$ (2.57) reduces to (2.55). In a two-class situation the condition $P_e \leq 0.5$ is always true. In this case therefore the two upper bounds coincide.

A lower bound on P_e is provided by the wellknown Fano [95] bound:

$$H \leq -P_e \ln P_e - (1-P_e)\ln(1-P_e) + P_e \ln(m-1) \qquad (2.59)$$

It can be shown [89] that there cannot be any tighter lower bound than (2.59).

Both (2.57) and (2.59) suffer from the difficulty that P_e cannot be solved in terms of H. But, still, error bounds can be obtained by considering ranges of possible values of H for different values of P_e. For a given value of P_e equalities in (2.57) and (2.59) correspond to H_{min} and H_{max}, respectively. In Fig. 2.8 the P_e bounds are illustrated by plotting H_{min} and H_{max} against P_e, subject to the condition that $P_e \leq 1 - 1/m$. Curves showing H_{max} values are plotted

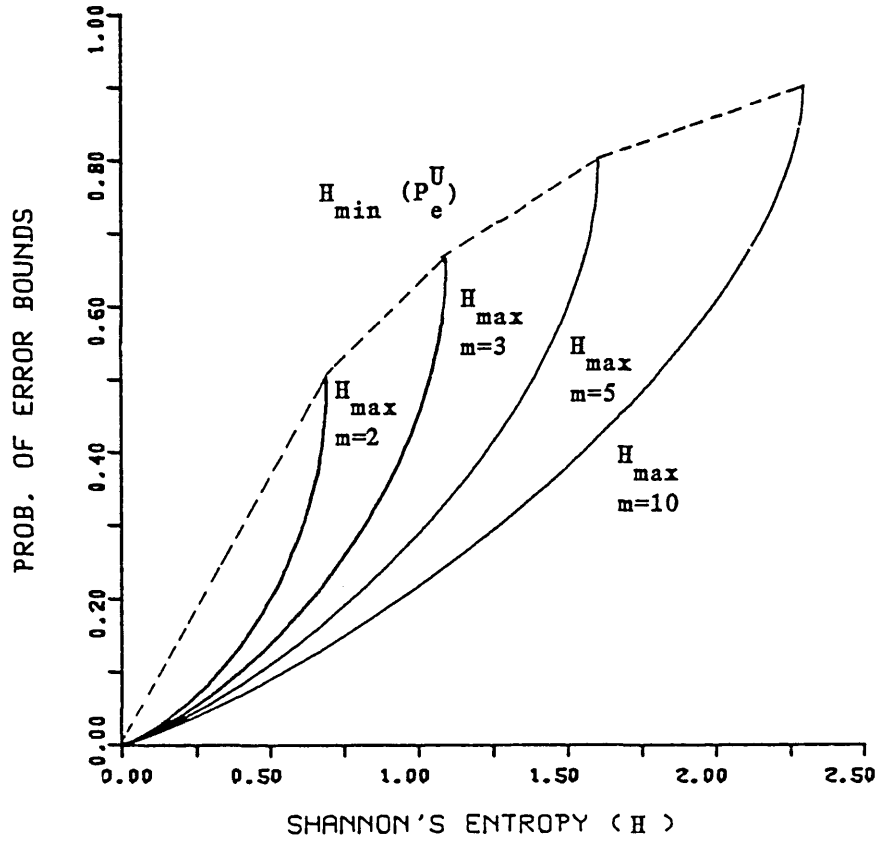


Fig. 2.8 Prob. of error (P_e) bounds versus Shannon's entropy (H)

for $m = 2,3,5$ and 10. H_{min} is shown by a dotted line in the diagram. As can be seen from the diagram, the Fano bound becomes loose with increasing m. This is due to the fact that $\ln(m-1)$ is a monotonically increasing function of m.

2.3.3 Mutual Information

The mutual information function [74] is defined by

$$I(X,C) = \sum_{i=1}^{m} \pi_i \int p(x|C_i) \ln \left[\frac{p(x|C_i)}{p(x)}\right] dx \qquad (2.60)$$

Mutual information can be expressed in terms of Shannon's conditional entropy and it can be shown that

$$I(X,C) = H(C) - H(C|X) \qquad (2.61)$$

where

$$H(C) = -\sum_{i=1}^{m} \pi_i \ln \pi_i \qquad (2.62)$$

H(C) is the a'priori entropy of the classes and H(C|X) is Shannon's conditional entropy as defined in (2.54). Thus, I(X,C) may be interpreted as the average decrease of entropy (or uncertainty) concerning C which results if X is observed. For $H(C) = H(C|X)$ one has $I(X,C) = 0$. This 'no information' situation indicates that the knowledge of X does not lead to reduction in the uncertainty regarding C. This simple interpretation has led to a widespread use of mutual

information as a feature evaluation criterion [5]-[7], [75]-[79].

For a particular pattern recognition problem the a'priori probabilities of the classes are generally fixed. If the a'priori probabilities are fixed then, from the point of view of pattern recognition, contribution of H(C) is unimportant and H(C|X) is the only factor that matters. In this case, therefore, I(X,C) has the same advantages and disadvantages as are associated with the Shannon's entropy measure.

2.3.4 Bayesian Distance of Devijver [49]

The Bayesian distance is defined to be

$$B = B(C|X) = E\left\{ \sum_{i=1}^{m} [\, P(C_i|X) \,]^2 \right\} \tag{2.63}$$

Devijver has derived the following inequalities between the Bayesian distance and the Bayesian probability of error:

$$\frac{1}{2}\,[\,1 - B\,] \leq 1 - \sqrt{B} \leq \frac{m-1}{m}\left[1 - \sqrt{\frac{mB - 1}{m - 1}}\,\right] \leq P_e \leq 1 - B \tag{2.64}$$

The difference between the upper bound and the tightest lower bound is

$$\delta = (1 - B) - \frac{m - 1}{m}\left[1 - \sqrt{\frac{mB - 1}{m - 1}}\right] \tag{2.65}$$

By differentiating with respect to B and equating the derivative to zero it is seen that the maximum value of δ occurs at

$$B = \frac{m + 3}{4\,m} \tag{2.66}$$

and the corresponding maximum δ-value is

$$\delta_{max} = \frac{1}{4} - \frac{1}{4\,m} \tag{2.67}$$

Therefore

$$\frac{1}{8} \leq \delta_{max} \leq \frac{1}{4} \tag{2.68}$$

Thus with increase in number of classes from 2 to ∞ the maximum difference between the two error bounds increases from 1/8 to 1/4.

It should be pointed out here that Devijver [49] also derived the results mentioned in (2.66), (2.67) and (2.68).

For $m = 2$ the Bayesian distance is related to the Ito's Q_r - function and to Lissack and Fu's K_α - function by the relationship

$$Q_0 = 1 - B = \frac{1}{2}[1 - K_2] \tag{2.69}$$

It is thus clear that the computational difficulties are the same for these measures.

In passing it may be noted that though the Bayesian distance does not assume a closed-form expression for the Gaussian distribution, it assumes [19] a closed-form expression for Laplacian-type distributions.

2.3.5 Conditional Quadratic Entropy of Vajda

The conditional quadratic entropy of Vajda [36] is defined to be

$$h = h(C|X) = E\left\{ \sum_{i=1}^{m} P(C_i|X) \; [\, 1 - P(C_i|X) \,] \right\} \tag{2.70}$$

This measure may be obtained from H by replacing $-\ln P(C_i|X)$ with $1 - P(C_i|X)$.

The probability of error bounds are given [19] by

$$\frac{h}{1 + \sqrt{1 - 2h}} \leq P_e \leq h \tag{2.71}$$

It is easy to see that

$$B = 1 - h \tag{2.72}$$

and thus, effectively, the Bayesian distance and the conditional quadratic entropy are two forms of the same measure.

2.3.6 Minkowski's Measures of Nonuniformity

Minkowski's measures of nonuniformity, proposed by Toussaint [43], are defined to be

$$M_k = M_k(C|X) = E\left\{ \sum_{i=1}^{m} \left| P(C_i|X) - \frac{1}{m} \right|^{\frac{2(k+1)}{2k+1}} \right\},$$

$$k = 0, 1, 2, \ldots \tag{2.73}$$

The following result proved by Toussaint provides P_e bounds for different values of k:

$$\left[\frac{\frac{m-1}{m} - P_e}{(\frac{1}{m})^{\frac{2}{r(k)}} - 1}\right]^{\frac{r(k)}{2}} \geq M_k \geq P_e \left[(\frac{m}{m-1})\, P_e - 2\right] + \frac{m-1}{m} \tag{2.74}$$

where

$$r(k) = \frac{2(k+1)}{2k+1} \tag{2.75}$$

The inequality in the left hand side of (2.74) yields the following P_e upper bound:

$$P_e \leq \frac{m-1}{m} - M_k^{\frac{2}{r(k)}} \left(\frac{1}{m}\right)^{\frac{2}{r(k)} - 1} \tag{2.76}$$

An analysis of (2.76) indicates that the bound becomes loose with an increase either in m or in k. It is again obvious from the right hand side inequality of (2.74) that the lower bound on P_e becomes loose with increasing m.

The tightest bounds are provided by M_0 which is related to the quadratic entropy of Vajda and to the Bayesian distance of Devijver. The relationship is

$$\frac{m-1}{m} - M_0 = h = 1 - B \qquad (2.77)$$

which implies that M_0 is as effective (or ineffective!) a feature evaluation criterion as are h and B.

2.4 Use of a Two-Class Measure in a Multiclass Problem

There are two approaches to solving a multiclass problem with the help of a two-class measure. These are known as the 'expected value' approach and the 'maximin' approach. In the first approach an average value of the 2-class measure is calculated from its values for different pairs of classes. The feature subset leading to maximum (or minimum, depending on the measure) average value is then selected for use in the recognition stage. In the second approach minimum values are considered instead of averages.

Ease of mathematical treatment has led to the preference of the expected value approach over the maximin approach. Results showing the relationships of the m-class probability of error with various 2-class measures have been established.

It is wellknown [96] that the error probability involved in an m-class problem is bounded above by the following inequality

$$P_e \leq \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} P_e(i,j) \tag{2.78}$$

where $P_e(i,j)$ is the pairwise error probability between the classes C_i and C_j. Using the above inequality Lainiotis [39] obtained a class of upper bounds on P_e, a particular case of which gives the bound in terms of the pairwise Bhattacharyya coefficients. This bound is given by

$$P_e \leq \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (\pi_i \pi_j)^{1/2} \rho_{ij} \tag{2.79}$$

Toussaint [68],[97],[98] obtained bounds to P_e in terms of the Kolmogorov variational distance and the divergence measures, and Lissack and Fu [58] derived error bounds in terms of the pairwise separability measure K_α (for $\alpha = 1$ and 2). Rather than going into the details of these bounds it is only necessary to comment here that, in their own right, the two-class measures have established their usefulness in the solution of an m-class problem.

2.5 Comparative Review of Measures

In the case of two-class measures the equations (2.13a) and (2.13b) show that the Bhattacharyya coefficient and the Matusita distance are basically two different versions of the same measure.

The inequality (2.23) suggests that the divergence function is less effective than the Bhattacharyya coefficient. From (2.10) and (2.22) it can be said that the Bhattacharyya distance is simpler to compute for the Gaussian processes with unequal covariances but equal means whereas, for the Gaussian processes with distinct means and equal covariances, the divergence function is easier to compute. The Kolmogorov variational distance has exactly the same computational difficulties as those of the probability of error.

For m = 2 one has,

$$Q_0 = 1 - B = h = \frac{1}{2} - M_0 = \frac{1}{2}\left[1 - K_2\right] \tag{2.80}$$

This relationship indicates that, in a two-class situation, Ito's Q_r function (for r = 0), the Bayesian distance, the conditional quadratic entropy, Minkowski's measure of nonuniformity M_k (for k = 0), and Lissack and Fu's separability measure K_α (for $\alpha = 2$) are equally effective measures. Though they provide tighter error bounds than the Bhattacharyya coefficient and the divergence function, they are difficult to compute. Toussaint's measure of affinity τ_r also suffers from the same difficulty. τ_0 and the measures mentioned in (2.80) have the computational advantage in that they can be estimated by classifying the sample data with the 1NN rule.

It was pointed out in section 2.1 that though both the upper and the lower bounds are required for approximating P_e, the upper bound is of more positive value as an aid to feature assessment. In the two-class situation an indication of the comparative effectiveness of

various measures may be obtained from the following inequality relating various upper bounds:

$$P_e = \frac{1}{2} - K \leq Q_r \leq \ldots \leq Q_0 = 1 - B = h = \frac{1}{2}\left[1 - K_2\right]$$

$$= \frac{1}{2} - M_0 \leq \frac{1}{2 \ln 2} H \leq \sqrt{\pi_1 \pi_2}\, \rho = \sqrt{\pi_1 \pi_2}\left(1 - \frac{\gamma^2}{2}\right) \quad (2.81)$$

For m classes ($m \geq 2$) the relationship given in (2.77) indicates that the conditional quadratic entropy of Vajda and the Bayesian distance of Devijver are in fact a special case of the Minkowski's measure of nonuniformity. For $m \geq 2$ the following inequality holds between various upper and lower bounds:

$$\frac{1}{2}(1 - B) \leq 1 - \sqrt{B} \leq \frac{h}{1 + \sqrt{1 - 2h}} \leq \frac{m - 1}{m}\left[1 - \sqrt{\frac{mB - 1}{m-1}}\right]$$

$$\leq P_e \leq h = 1 - B = \frac{m - 1}{m} - M_0 \leq \frac{1}{2 \ln 2} H \quad (2.82)$$

The ordering of the different measures present in the above inequality, from the point of view of accuracy, is self-explanatory apart from the fact that because of the exact relationships between some measures the bounds in terms of one may be expressed in terms of the others.

It is difficult to give a comparative picture of the computational complexities of the various multiclass measures. Closed-form expressions for most of the measures do not exist, rendering them difficult to compute. In the nonparametric case usually the computation of a probabilistic distance measure has to be

preceded by the estimation of the class-conditional probability density functions. Probability density function estimation itself is a computationally complex task. Compared to this the differences in the computational complexities of various multiclass measures are quite small. This makes the comparison of the computational complexities less important.

From the above analysis and discussions it may be noted that the Bayesian distance (and the measures exactly related to it) has tighter error bounds than most of the other measures. But even with this measure, and for $m = 2$, the difference between the upper and the lower error bounds can be as large as 0.125. This difference increases with increasing m and approaches 0.250 as m tends to ∞, which, in practice, is likely to be too large. Thus, even with a 'difficult to compute' measure like the Bayesian distance the accuracy of the method is less than satisfactory. Therefore, with the exception of certain parametric cases where some measures assume closed-form expressions, it may be better to estimate the error by a direct method instead of employing the computation of the probabilistic measures, unless, of course, an easy method exists for estimating the probabilistic measures.

CHAPTER 3

NEW MAHALANOBIS DISTANCE-BASED FEATURE EVALUATION CRITERIA

3.1 Introduction

Information on the probability structure is a prerequisite for obtaining the probabilistic separability measures. Computation of these measures usually involves integration of multivariate density functions which themselves have to be estimated from the training data. Computational complexity increases exponentially with increases in the dimension of the feature vector. Moreover, in a real-life situation the size of the training set is sometimes too small for estimation of higher order interactions of the features because one would then face the wellknown 'curse of dimensionality' problem [99].

In the past, researchers quite often assumed the independence of features in order to implement various feature evaluation measures. This was probably because they wanted to avoid the estimation and the computational needs involved in a higher dimensional analysis. With the decreasing cost of data storage and the increasing computational speeds of the present day computers it is sometimes worthwhile to take into account the higher order interactions of the features. The Mahalanobis distance is a simple measure which takes into account the effects of correlations between the features. This, together with the fact that it provides a distribution-free upper bound to P_e, indicates the potentiality of the Mahalanobis distance as a feature evaluation criterion. Though the application of the Mahalanobis distance in pattern recognition is not new, it deserves further attention.

In the present chapter certain theoretical properties of the Mahalanobis distance are investigated which are relevant in the context of feature evaluation. The Mahalanobis distance is defined first. Its relationships with the Bayesian probability of error are then discussed, both in the distribution-free case and in the case when the feature vector follows a Gaussian distribution. Sample-based Mahalanobis distance is then described. This is followed by the development of two functions of the Mahalanobis distance and it is proposed that they be used as feature evaluation criteria. As will be shown later in this chapter, these two new criteria can have values in the range 0 to 1. It is expected that, in a muliclass situation, this boundedness will result in their better performance as compared with the direct use of the Mahalanobis distance. In the present chapter discussions on some Mahalanobis distance-based statistics are included which have their origin in the statistical theory of testing of hypothesis and are thought to be useful in feature evaluation in the context of pattern recognition.

3.2 Definition and Properties of the Mahalanobis Distance

3.2.1 Definition

Mahalanobis distance between two classes C_1 and C_2 is defined by

$$\Delta^2 = (\mu_1 - \mu_2)' \, V^{-1} \, (\mu_1 - \mu_2) \tag{3.1}$$

where

$$V = \pi_1 V_1 + \pi_2 V_2 \tag{3.2}$$

where μ_1 and μ_2 are the mean vectors and V_1 and V_2 are the dispersion matrices of X in C_1 and C_2, and π_1 and π_2 are the a'priori probabilities of the two classes.

It should be noted here that Δ^2 is a generalization [100] of the original distance function of Mahalanobis [101] which was defined as the distance between two Gaussian distributions with a common dispersion matrix V. The only assumption required is that of nonsingularity of V defined in (3.2).

3.2.2 Relationships Between Δ^2 and P_e

3.2.2.1 Distribution-free relationship

In the distribution-free case there cannot be any exact relationship between Δ^2 and P_e. In this case, however, P_e is upper bounded by the following relationship:

$$P_e \leq \frac{2 \pi_1 \pi_2}{1 + \pi_1 \pi_2 \Delta^2} \tag{3.3}$$

For a proof of the above result reference may be made to chapters 2 and 4 of the book by Devijver and Kittler [100].

For equal a'priori probabilities of the classes the inequality (3.3) reduces to

$$P_e \leq \frac{2}{4 + \Delta^2} \tag{3.4}$$

It may be noted that the upper bound (3.4) holds good for any values of the a'priori prababilities. Though (3.4) is a loose bound it is useful when the a'priori probabilities are either unknown or equal. In Fig. 3.1 a diagrammatic representation of the upper bound (3.3) is given for π_1 = 0.500, 0.625, 0.750 and 0.875.

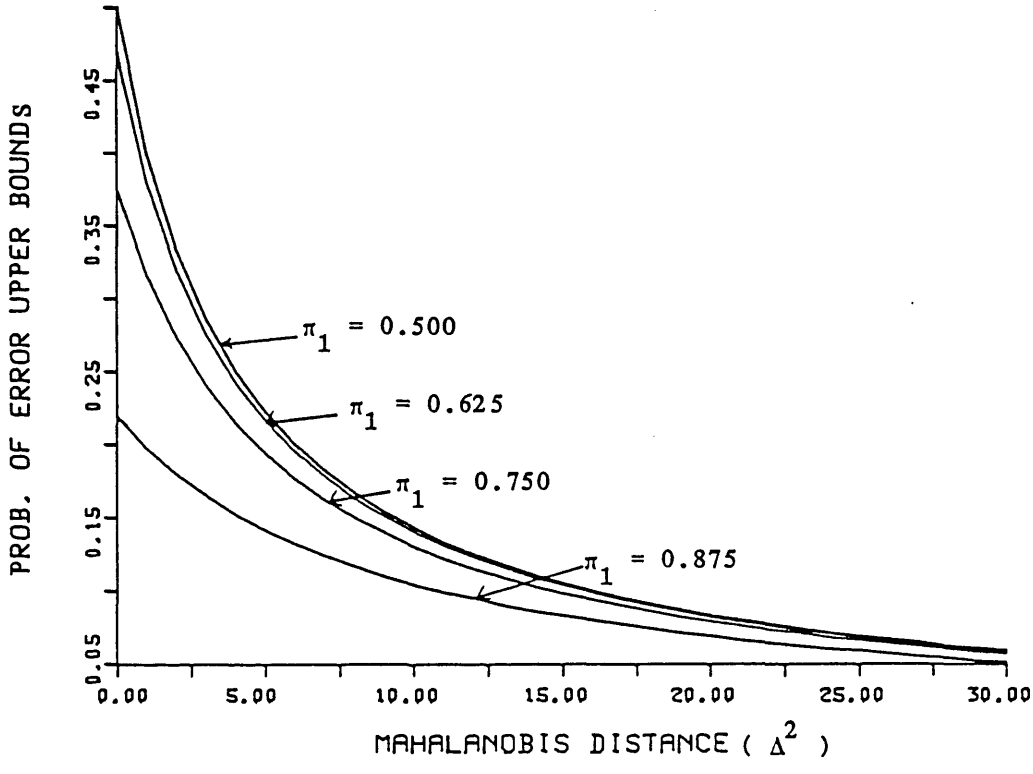


Fig. 3.1 Distribution-free prob. of error (P_e) upper bounds in terms of the Mahalanobis distance (Δ^2) for different a'priori probabilities

3.2.2.2 Relationship for Gaussianly-Distributed Features

If X follows a Gaussian distribution with a common dispersion matrix V for the two classes C_1 and C_2, then the relationship between Δ^2 and P_e is exact. This exact relatioship is given by the following wellknown [35] equation:

$$P_e = \pi_1 \Phi\left(\frac{\alpha - \frac{1}{2}\Delta^2}{\Delta}\right) + \pi_2 \left[1 - \Phi\left(\frac{\alpha + \frac{1}{2}\Delta^2}{\Delta}\right) \right] \tag{3.5}$$

where

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{t^2}{2}} dt \tag{3.6}$$

and

$$\alpha = \ln\left(\frac{\pi_1}{\pi_2}\right) \tag{3.7}$$

For $\pi_1 = \pi_2$ the equation (3.5) reduces to

$$P_e = \Phi\left(-\frac{1}{2}\Delta\right) \tag{3.8}$$

The relationship (3.8) is illustrated in Fig. 3.2. It is observed that P_e, a monotonically decreasing function of Δ^2, becomes less than 5% for $\Delta^2 = 11$ and becomes less than 1% for $\Delta^2 = 22$. In a later section (section 3.3.3.2) discussion will be made of the question of how well the existing P_e upper bounds, expressed in terms of Δ^2, compare with the above exact relationship.

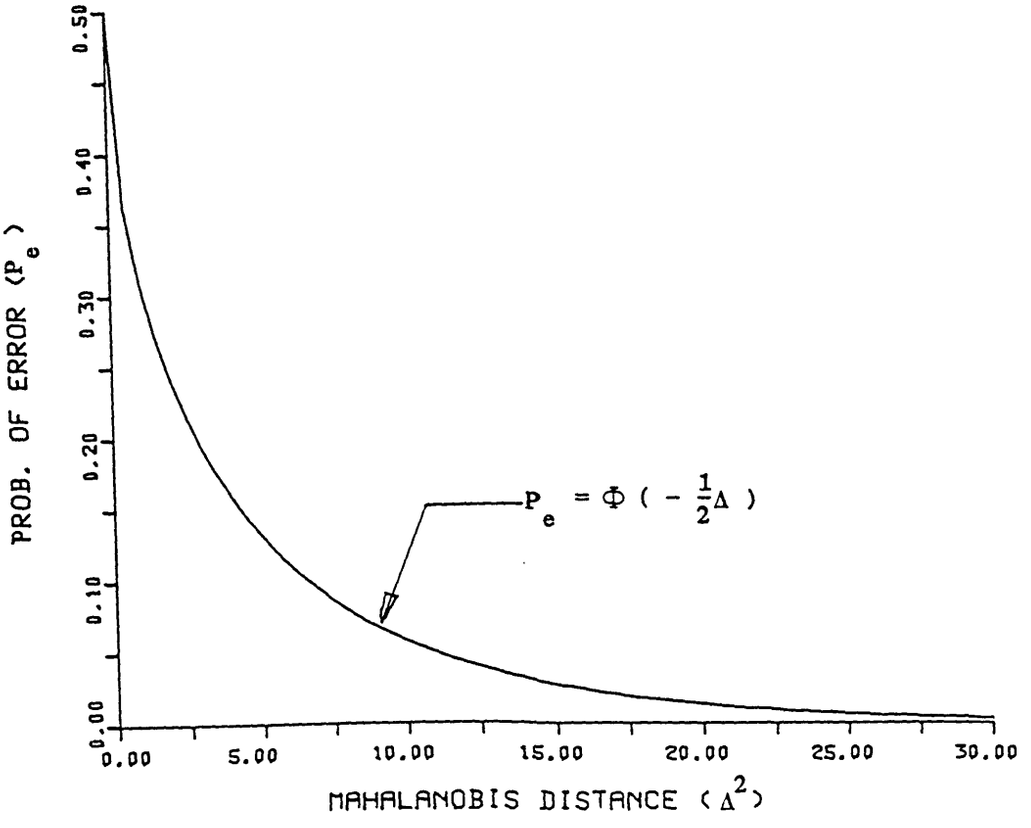


Fig.3.2 Exact relationship between the Mahalanobis distance (Δ^2) and the prob. of error (P_e) in the Gaussian case with common dispersion matrix and equal a'priori probabilities

3.2.3 Mahalanobis Distance as a Special Case of the Divergence Function

As mentioned in section 2.2.3 of the previous chapter, in the case of the Gaussian feature vector the divergence function takes the closed-form expression given by (2.22). If it is further assumed that the two dispersion matrices V_1 and V_2 are equal then (2.22) reduces to (3.1) confirming the wellknown [102] fact that the Mahalanobis distance, in its restricted sense of definition, is a particular case of the divergence function.

3.2.4 Sample-Based Mahalanobis Distance

The Mahalanobis distance Δ^2 is defined in terms of the population parameters μ_1, μ_2, V_1 and V_2. In real-life problems these population parameters are usually not available and are, therefore, estimated from the sample data. Replacing the population parameters by the corresponding sample statistics in (3.1) the following sample-based Mahalanobis distance is obtained:

$$D^2 = (\bar{x}_1 - \bar{x}_2)' \, S^{-1} \, (\bar{x}_1 - \bar{x}_2) \tag{3.9}$$

where

$$S = \frac{(N_1 - 1) S_1 + (N_2 - 1) S_2}{N_1 + N_2 - 2} \tag{3.10}$$

$$\bar{x}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} x_{1i} \tag{3.11}$$

$$\bar{x}_2 = \frac{1}{N_2} \sum_{i=1}^{N_2} x_{2i} \tag{3.12}$$

$$S_1 = \frac{1}{N_1 - 1} \sum_{i=1}^{N_1} (x_{1i} - \bar{x}_1)(x_{1i} - \bar{x}_1)' \tag{3.13}$$

$$S_2 = \frac{1}{N_2 - 1} \sum_{i=1}^{N_2} (x_{2i} - \bar{x}_2)(x_{2i} - \bar{x}_2)' \tag{3.14}$$

and where $(x_{11}, x_{12}, \ldots, x_{1N_1})$ and $(x_{21}, x_{22}, \ldots, x_{2N_2})$ are the samples of sizes N_1 and N_2 from the classes C_1 and C_2, respectively. $\bar{x}_1$ and $\bar{x}_2$ are unbiassed estimates of μ_1 and μ_2, respectively. S is unbiassed for V under the assumption of common dispersion matrix for the two classes.

3.3 Two New Mahalanobis Distance–Based Feature Evaluation Criteria

3.3.1 Introduction

The Mahalanobis distance Δ^2 takes on values in the range of 0 to ∞, the higher values representing greater separability of the class means. This unbounded increase in the value of Δ^2 with increasing separability is not unwelcome in a two-class pattern recognition problem because it does not pose any difficulty in the comparison of different feature sets. But, the unboundedness poses a serious difficulty when the expected value approach is adopted and the Mahalanobis distance (a two-class measure) is applied in a multiclass problem. Just one exceptionally large value of Δ^2 in the set of ${}^{m}C_{2}$ values would lead to a high value of the average which would then fail to represent the average separability of the m classes. One way to reduce this drawback would be to transform the Δ^2-values, before averaging them, in such a way that the transformed measure lies within a finite range. In the present section two such transformations on Δ^2 are obtained. They are given below:

$$\Delta_A^2 = \frac{\pi_1 \pi_2 \Delta^2}{1 + \pi_1 \pi_2 \Delta^2} \tag{3.15}$$

$$\Delta_B^2 = 1 - \exp\left(-\frac{\Delta^2}{8}\right) \tag{3.16}$$

In the next two subsections (subsections 3.3.2 and 3.3.3) the derivations and some useful properties associated with the new measures will be described.

3.3.2 Derivations of Δ_A^2 and Δ_B^2

3.3.2.1 Derivation of Δ_A^2

The inequality (3.3) gives a distribution-free upper bound of P_e expressed in terms of Δ^2 (it is believed that this is the only distribution-free upper bound available in terms of Δ^2). Let this upper bound be denoted by $P_e^U(A)$ as follows:

$$P_e^U(A) = \frac{2\,\pi_1\,\pi_2}{1 + \pi_1\,\pi_2\,\Delta^2} \tag{3.17}$$

It is easy to see that

$$P_e^U(A) \;\varepsilon\; [0,\; 2\pi_1\pi_2]$$

To make the range of values of $P_e^U(A)$ independent of π_1 and π_2 divide $P_e^U(A)$ by $2\pi_1\pi_2$. Now

$$\frac{P_e^U(A)}{2\pi_1\pi_2} \;\varepsilon\; [0\;,\;1]$$

and as Δ^2 increases from 0 to ∞ the value of $P_e^U(A)\;/\;2\pi_1\pi_2$ decreases from 1 to 0. To bring the changes in the same direction $P_e^U(A)\;/\;2\pi_1\pi_2$ is subtracted from 1. Call it Δ_A^2. Thus,

$$\Delta_A^2 = 1 - \frac{P_e^U(A)}{2\pi_1\pi_2} = \frac{\pi_1\pi_2\Delta^2}{1 + \pi_1\pi_2\Delta^2}$$

such that $\Delta_A^2 \;\varepsilon\; [0,1]$. In other words, it is a normalizing transformation on $\Delta^2 \;\varepsilon\; [0,\infty)$. This completes the derivation of Δ_A^2 defined in (3.15).

3.3.2.2 Derivation of Δ_B^2

An upper bound of P_e expressed in terms of the Bhattacharyya coefficient, ρ, is given by

$$P_e \leq \sqrt{\pi_1 \pi_2}\ \rho \tag{3.18}$$

In the case of Gaussian distributions with a common dispersion matrix the Mahalanobis distance and the Bhattacharyya coefficient are related as follows:

$$\Delta^2/8 = -\ \ln\ \rho \tag{3.19}$$

In this case, therefore, the upper bound mentioned above can be written as

$$P_e \leq \sqrt{\pi_1 \pi_2}\ \exp(-\ \Delta^2/8) \tag{3.20}$$

Let this upper bound be denoted by $P_e^U(B)$. It is easy to see that

$$P_e^U(B) \in [0, \sqrt{\pi_1 \pi_2}]$$

Therefore

$$\frac{P_e^U(B)}{\sqrt{\pi_1 \pi_2}} \in [0 , 1]$$

Following the same arguments as in the case of Δ_A^2 leads to the criterion

$$\Delta_B^2 = 1 - \frac{P_e^U(B)}{\sqrt{\pi_1 \pi_2}} = 1 - \exp\left(- \frac{\Delta^2}{8} \right)$$

which can have values in the range [0,1]. Thus the derivation of Δ_B^2 is completed.

3.3.3 Properties of Δ_A^2 and Δ_B^2

3.3.3.1 Boundedness and Monotonicity

As is obvious from the derivations of Δ_A^2 and Δ_B^2, both of them are bounded by [0,1] whereas the range of Δ^2 is $[0,\infty)$. Usefulness of this boundedness in an m-class situation has already been discussed in the section 3.3.1. Both Δ_A^2 and Δ_B^2 are monotonically increasing functions of Δ^2. Their relationships with Δ^2 are illustrated diagrammatically in Fig. 3.3. It is observed that Δ_B^2 approaches the upper limit of 1.0 faster than Δ_A^2.

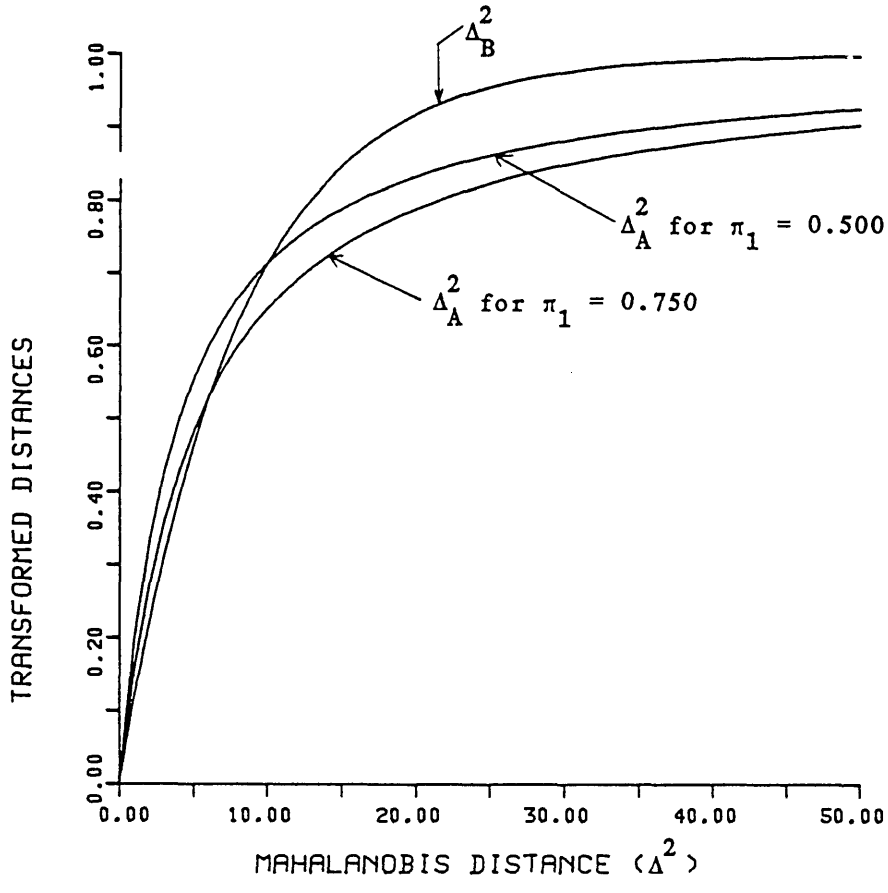


Fig. 3.3 Showing Δ_A^2 (for π_1 = 0.50 and 0.75) and Δ_B^2 in terms of Δ^2

3.3.3.2 Relationships with P_e

Since Δ_A^2 and Δ_B^2 are such functions of the Mahalanobis distance Δ^2 that Δ^2 can be solved in terms of them, the distribution-free upper bound (3.3) and the upper bound in the Gaussian case (3.20), which are expressed in terms of Δ^2, can also be expressed in terms of both Δ_A^2 and Δ_B^2. This implies that the error bounds associated with Δ_A^2 and Δ_B^2 are the same as those associated with Δ^2. Strictly speaking, therefore, in a two-class situation one cannot differentiate between Δ_A^2 and Δ_B^2 from the point of view of tightness of P_e bounds provided by them. In view of the functional behaviours of Δ_A^2 and Δ_B^2 illustrated in Fig. 3.3, in a multiclass situation they would lead to different approximations to P_e, because in this case, for each of the two criteria, one has to combine the criterion values for different pairs of classes. In absence of useful multiclass bounds one has to depend on the experimental results. The experimental findings will be discussed in chapter 6.

It may be noted that Δ_A^2 and Δ_B^2 have been developed based on the P_e upper bounds given by the inequalities (3.3) and (3.20), respectively. Therefore, a comparison of these two upper bounds would give some insight into the comparative effectiveness of Δ_A^2 and Δ_B^2 as feature evaluation criteria. In Fig. 3.4 the P_e bounds given by (3.3) and (3.20) are plotted for $\pi_1 = \pi_2 = 1/2$. As a reference for comparison (3.8) is also plotted which gives the exact value of P_e as a function of Δ^2 for Gaussian distributions with common dispersion matrix. It is observed that for $\Delta^2 \leq 10$ (roughly) the distribution-free upper bound ($P_e^U(A)$) gives a closer approximation than the bound corresponding to Δ_B^2 ($P_e^U(B)$) and for $\Delta^2 > 10$ the

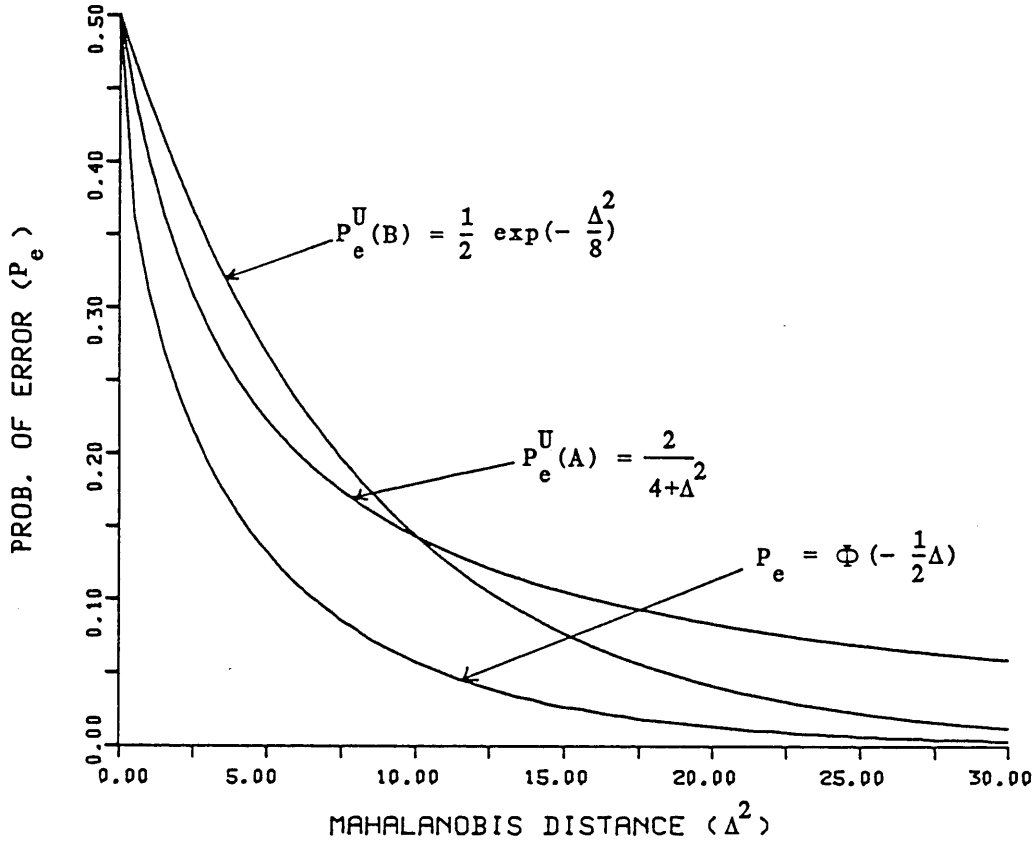


Fig. 3.4 Prob. of error (P_e) versus Mahalanobis distance (Δ^2) for equal a'priori probabilities of the two classes

latter approaches P_e faster than the earlier. Taking into consideration the fact that $P_e^U(A)$ is a distribution-free bound, from Fig. 3.4 it may be commented that $P_e^U(A)$ is a reasonably good approximation to P_e. For $\Delta^2 > 10$, $P_e^U(B)$ provides tighter bound but it is more difficult to compute than $P_e^U(A)$. Moreover, unlike $P_e^U(A)$, $P_e^U(B)$ is not a distribution-free bound. Therefore the criterion Δ_A^2 seems to have a preference over Δ_B^2.

As mentioned earlier, as far as the multiclass pattern recognition problem is concerned, decision as to which of the criteria to select will ultimately depend on their experimental performance.

3.3.3.3 Δ_B^2 and J_T: An Observation

It is an interesting coincidence that the functional structures of the Δ_B^2 criterion and the transformed divergence function J_T of Swain, *etal*. [56],[57] are very much similar to each other. For the ease of comparison they are written together below:

$$\Delta_B^2 = 1 - \exp(-\Delta^2/8)$$

$$J_T = 2[\ 1 - \exp(-J/8)]$$

As mentioned earlier, in the case of Gaussian distributions with common dispersion matrix the divergence function J reduces to the Mahalanobis distance function Δ^2. In this case, therefore, the only difference between Δ_B^2 and J_T would be the presence of the multiplying constant 2 in the expression for J_T.

3.3.4 Sample Analogues of Δ_A^2 and Δ_B^2

Replacing Δ^2 by D^2 in the expressions for Δ_A^2 and Δ_B^2 the following sample analogues of them are obtained:

$$D_A^2 = \frac{\pi_1 \pi_2 D^2}{1 + \pi_1 \pi_2 D^2} \tag{3.21}$$

and

$$D_B^2 = 1 - \exp(-D^2/8) \tag{3.22}$$

where D^2 is the sample-based Mahalanobis distance defined in (3.9).

3.4 Use of Some Existing Mahalanobis Distance-Based Statistics in Feature Evaluation

3.4.1 Two D^2-Based Statistics

Two D^2-based statistics, one for testing the between-class differences and the other for testing the sufficiency of a subset of 'features' (usually known as 'variables' in the Statistics literature), have been used in the past as a means of evaluating the performance of a discriminant function. In the present section (section 3.4) they will first be described and then the possibility of their use as feature evaluation criteria will be looked into. It is worth pointing out here that for the purpose of the present section the knowledge of the a'priori probabilities will be assumed to be

absent and the sample dispersion matrix S will be estimated using the equation (3.10).

3.4.2 Test of Between-Class Differences

For a given set of features the objective of this test is to decide whether the differences between the two classes are significant. The test statistic is defined [103] to be

$$F_1 = \frac{N_1 N_2 (N_1 + N_2 - n - 1)}{(N_1 + N_2) (N_1 + N_2 - 2) n} . D^2 \qquad (3.23)$$

where N_1 and N_2 are the sizes of the samples from the two classes, n is the number of features under consideration and D^2 is the sample-based Mahalanobis distance as defined in (3.9).

The test statistic F_1 follows F-distribution with (n, N_1+N_2-n-1) degrees of freedom for data coming from Gaussian distributions with common dispersion matrix. It is actually a test statistic for testing the equality of the two mean vectors. The test determines if the present set of features has 'significant' discriminating ability to ensure classification of future observations.

3.4.3 Test of Sufficiency of a Subset of Features

Partition $X = (X_1, X_2, \ldots, X_n)'$ into $X_1 = (X_1, X_2, \ldots, X_r)'$ and $X_2 = (X_{r+1}, \ldots, X_n)'$. Test of sufficiency of X_1 for discrimination between C_1 and C_2 is made using the following test statistic [104]:

$$F_2 = \frac{N_1 + N_2 - n - 1}{n - r} \cdot \frac{C (D_n^2 - D_r^2)}{1 + C.D_r^2} \tag{3.24}$$

where D_n^2 and D_r^2 are the Mahalanobis D^2-statistics on the full set and the subset, respectively,

and

$$C = \frac{N_1 N_2}{(N_1 + N_2) (N_1 + N_2 - 2)} \tag{3.25}$$

F_2 follows F-distribution with n-r and N_1+N_2-n-1 degrees of freedom for Gaussian distributions with common dispersion matrix.

For r = n-1 the test statistic F_2 indicates if a single specified feature has any discriminating power. In this case F_2 reduces to

$$F_2' = \frac{N_1 + N_2 - n - 1}{1} \cdot \frac{C (D_n^2 - D_{n-1}^2)}{1 + C.D_{n-1}^2} \tag{3.26}$$

F_2' gives a measure of usefulness of the nth feature over (n-1) features.

3.4.4 Use of D^2-Based Statistics in Feature Evaluation

From the above discussion it is apparent that F_1, F_2 and F_2' can be used in feature evaluation. F_1 is a measure of the discriminatory ability of a set of n features. F_2 gives the amount of decrease in the discriminatory ability when a subset of r features are discarded from the full set of n features. F_2', which is a special case of F_2, gives a measure of effectiveness of one (nth) feature. This criterion can, therefore, be used in successive inclusion or deletion of features.

Since the criteria F_1 and F_2 take into account the sample sizes, they are expected to perform well as feature evaluation criteria in small sample situations. This is, however, subject to experimental verification because, for the non-Gaussian data, the criteria F_1 and F_2 will not have the nice distributional properties mentioned above.

CHAPTER 4

CHARACTER RECOGNITION: PREPROCESSING AND FEATURE EXTRACTION

4.1 Introduction

The computer recognition of characters has been a challenging problem to the researchers since the reporting of some early works in this field in late 1950's [1], [105]-[108]. Evergrowing interest in this field has been reflected through the publication of a large number of papers, books and special issues of journals, devoted sclely to character recognition, during the last twenty five years. For up-to-date information on the state of advancement in this field reference may be made to the survey papers [109], [110] and [111].

In the present chapter and the next two chapters various feature evaluation criteria are applied to the problem of the recognition of isolated handprinted numeric characters. As mentioned in the introductory chapter (chapter 1), the purpose of this experimental study is to compare the existing and the proposed featu-e evaluation criteria, and the development of a character recognition system is not of prime concern.

This chapter deals with the preprocessing of data which leads to the representation of a numeral by a set of features. In chapters 5 and 6 the feature ordering experiments and the recognition experiments are performed. These experiments will lead to a comparative assessment cf the feature evaluation criteria.

In section 4.2 of the present chapter the data set used in the study is described. Section 4.3 deals with the representation of data which consists of the processes of digitization, binarization and noise reduction. Application of these three processes on a numeric character leads to its representation by a matrix of '0's and '1's, with '1's representing the contour of the character and '0's representing the background. Let this matrix be called a 'character matrix'. The size of this matrix is determined by the resolution used during the digitization process. In section 4.4 two sets of features are extracted from the character matrix obtained in section 4.3. For reasons mentioned later (in section 4.4) these two sets of features are designated as the 'normalized frequency' features and the 'normalized chararcteristic loci' features. In section 4.5 a preliminary analysis of the data is provided for the detection of redundant features. These redundant features are excluded from further analysis to follow in chapters 5 and 6.

4.2 Data set

The data set consisted of 1000 isolated handprinted numerals written by 10 members of the Communication Section of the Department of Electrical Engineering, Imperial College of Science and Technology, London. Each member wrote 10 repetitions of each of the 10 numerals 0,1,2,...,9. The writing was done on transparent sheets using a black inked felt pen. The only restriction imposed on writing was to put each character in a square of size 12mm x 12mm. To facilitate this a

white paper with square boxes of size 12mm x 12mm drawn on it was put underneath the transparency as a guide. A copy of the data set is given in Appendix A.

4.3 Representation of Data

4.3.1 Digitization and Unpacking

The data were digitized by using a SCANDIG 3 scanner controlled by a NOVA 3 computer with magnetic tape facility, the system being installed in the Biophysics Department of the Imperial College. SCANDIG 3 is a powerful instrument with scan increment options of 25, 50, 100 and 200 microns and density resolution of 1 part in 256. Keeping in mind the storage requirements it was decided to select the scan increment option of 200 microns. This meant that each square of size 12mm x 12mm, within which a numeral was written, would be represented by a matrix of 60 pixels x 60 pixels, each of the pixels assuming a value in the grey level range of 0 to 255, the lower grey levels corresponding to the background pixels and the higher grey levels corresponding to the path of the character.

It is worthwhile mentioning that the characters were not digitized individually. As can be seen from Appendix A, each page of data contained 100 characters written within a square of size 120 mm x 120mm. This whole square was scanned as one image and stored on magnetic tape in 'packed' form. Thus, at the end of the scanning process the data set consisted of 10 digitized images stored on a

magnetic tape. For further analysis of the data the Imperial College computers Cyber 174 and Cyber 855 were used. To obtain the matrix representations of the individual characters, and for the convenience of further data handling to follow, the unpacking program listed in Appendix B 1 was used. This program took care of the compatibility requirements of a magnetic tape created on NOVA 3 but analyzed on the Cyber computers.

To summarize, at the end of the processes of digitization and unpacking, it was possible to represent each of the 1000 characters by a matrix of order 60 x 60 wherein each pixel could assume a value in the range of 0 to 255.

The digitized grey-tone representations of the character samples were identified by Z0011, Z0012, ..., Z0110; Z1011, ..., Z1110; Z9011, ..., Z9110.

4.3.2 Binarization

Though the character data under consideration were basically binary in nature, that is, black characters on a white background, due to differences in 'whiteness' and 'blackness' of different parts of an image the pixels could assume values in the range of 0 to 255. Unlike some image processing problems, for the purpose of character recognition it was not necessary to have the detailed grey level differences of the pixels. Binary representation was thought to be ideal.

Conversion from the multi-level representation (grey-tone image) to the binary representation (two-tone image) of a character matrix required a grey level threshold to be chosen which would enable one to decide whether a pixel belonged to the background or to the character locus. The choice of this grey level threshold was made on the basis of an analysis of the frequency distributions of grey levels of pixels in the character matrices. Though the possible range of grey level values was from 0 to 255, hardly any grey level value greater than 60 occured. This was probably due to some prior biasing in the scanner setting. However, it was not difficult to choose an acceptable threshold. As expected, bimodality was observed in the frequency distributions. Inspection of the frequency distributions suggested that the grey level values of 7 or less could be considered as belonging to the background. The threshold value of 7 was further justified by binarizing a few character matrices with different threshold values and visually comparing the binary images with the original writings. As an illustration of the binarization process, in Fig. 4.1a and Fig. 4.1b the grey-tone and the two-tone representations of the numeral '8' (encircled with a square in the Appendix A) are given.

A program for the grey-tone to two-tone conversion of character matrices is given in Appendix B.2. The actual conversion process takes place in the subroutine. The program is written in such a way that it can deal with more than one character matrix.

The binarized character samples were identified by B0011, B0012,

Z8033

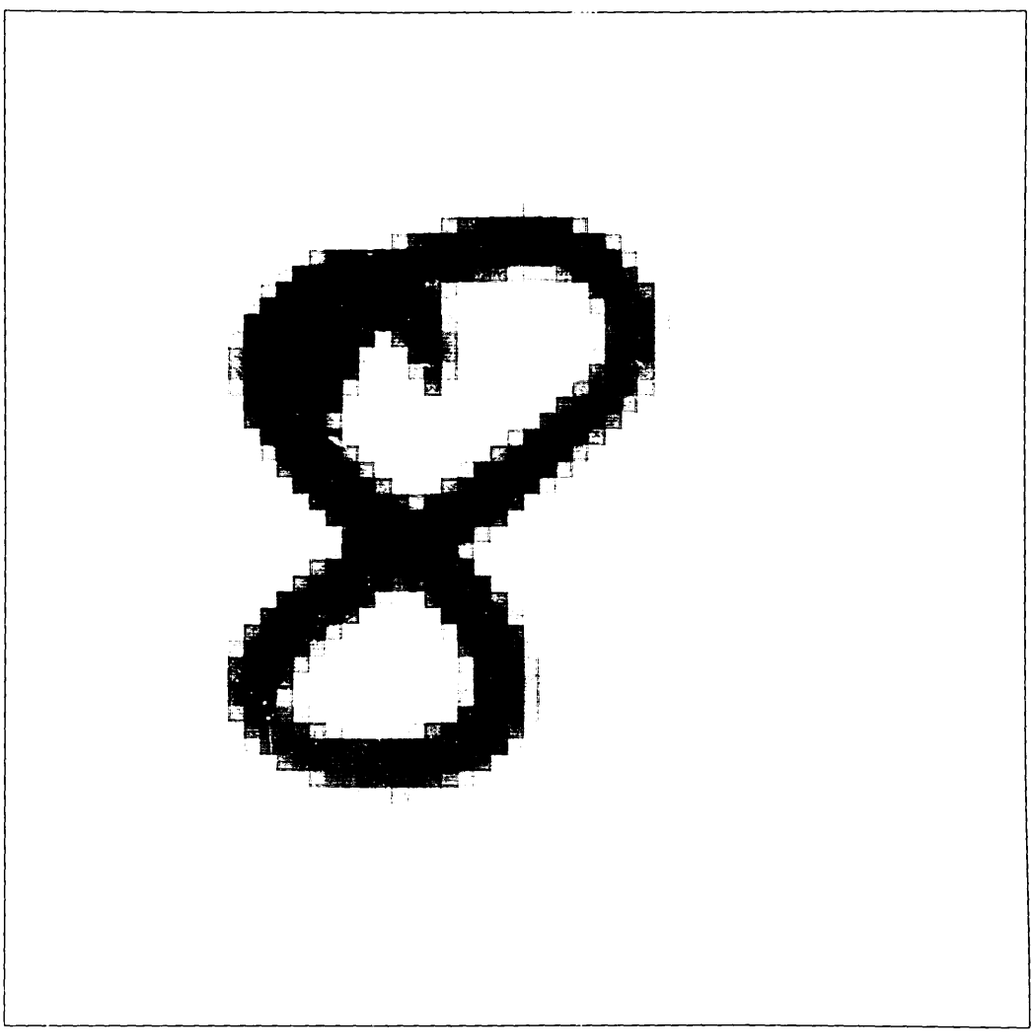


Fig. 4.1a Grey-tone representation of a numeral '8'

B8033

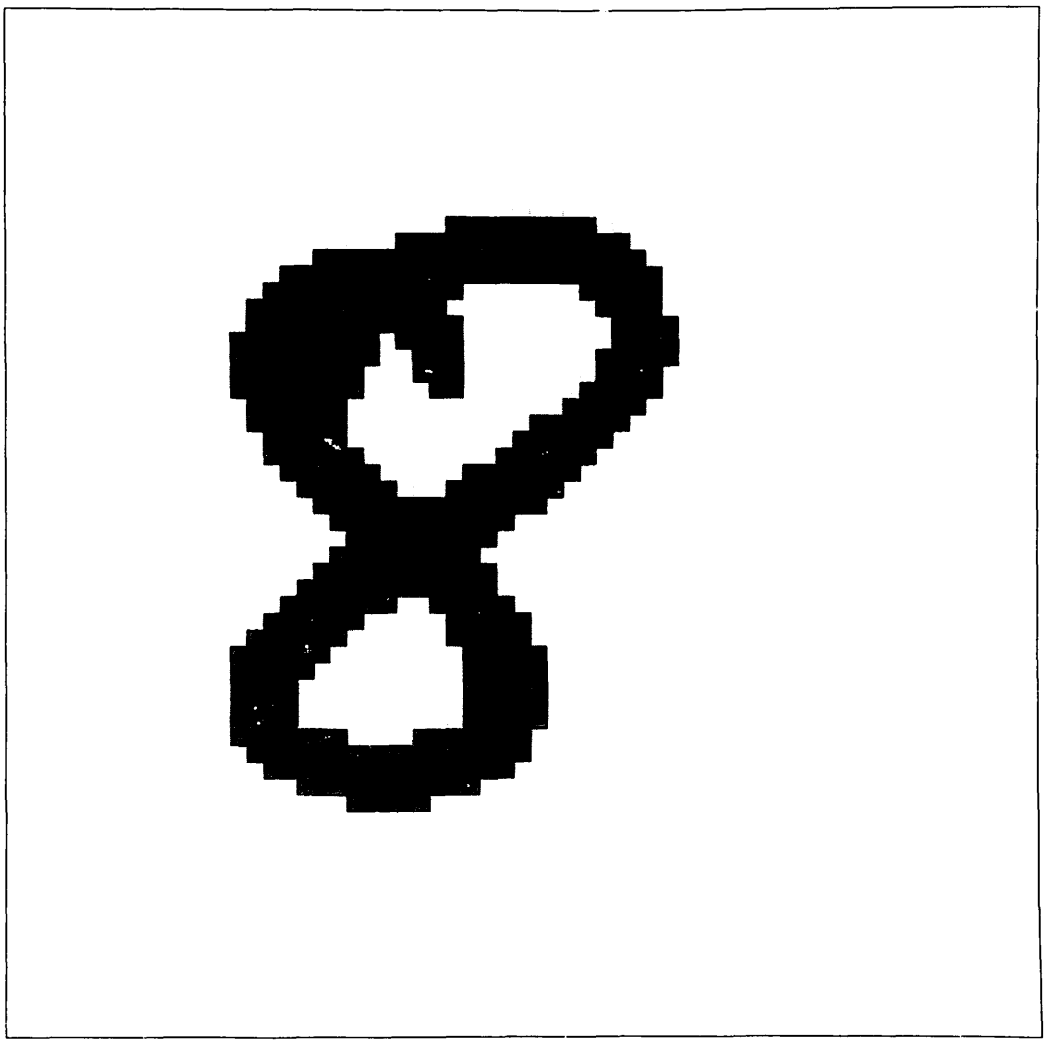


Fig. 4.1b Two-tone (binary) representation of the numeral of Fig. 4.1a

..., B0110; B1011, ..., B1110; ..., B9011, ..., B9110.

4.3.3 Noise Reduction: A Heuristic Scheme

Binarization error and spattering of ink sometimes lead to the presence of isolated blocks of pixels of value '1' (black pixels) in the background, or contain '0'-pixels which should ideally be '1', in the binary representation of a character matrix (for illustration see Figures 4.2a and 4.2b).

Various smoothing techniques have been used in the past [1], [112], [113] in order to reduce the 'noisy' pixels. In most of these techniques a 3 x 3 window has been considered and some logical (or averaging) rules applied to the pixel appearing in the middle of the window. From a visual inspection of the binarized characters under consideration it was apparent that a 3 x 3 window was not good enough to eliminate the noisy '1's. It was also observed that the occurence of noisy '1's was rather low in the character matrices. Moreover, the features to be considered will not be as greatly affected by the presence of noisy '0's as by the presence of the noisy '1's. It was, therefore, decided to develop a noise reduction algorithm to remove the isolated blocks of noisy '1's. An algorithm based on windows of different sizes varying from 3 x 3 to 5 x 5 was developed. Consideration of higher order windows enabled the elimination of bigger noise specks.

B3045

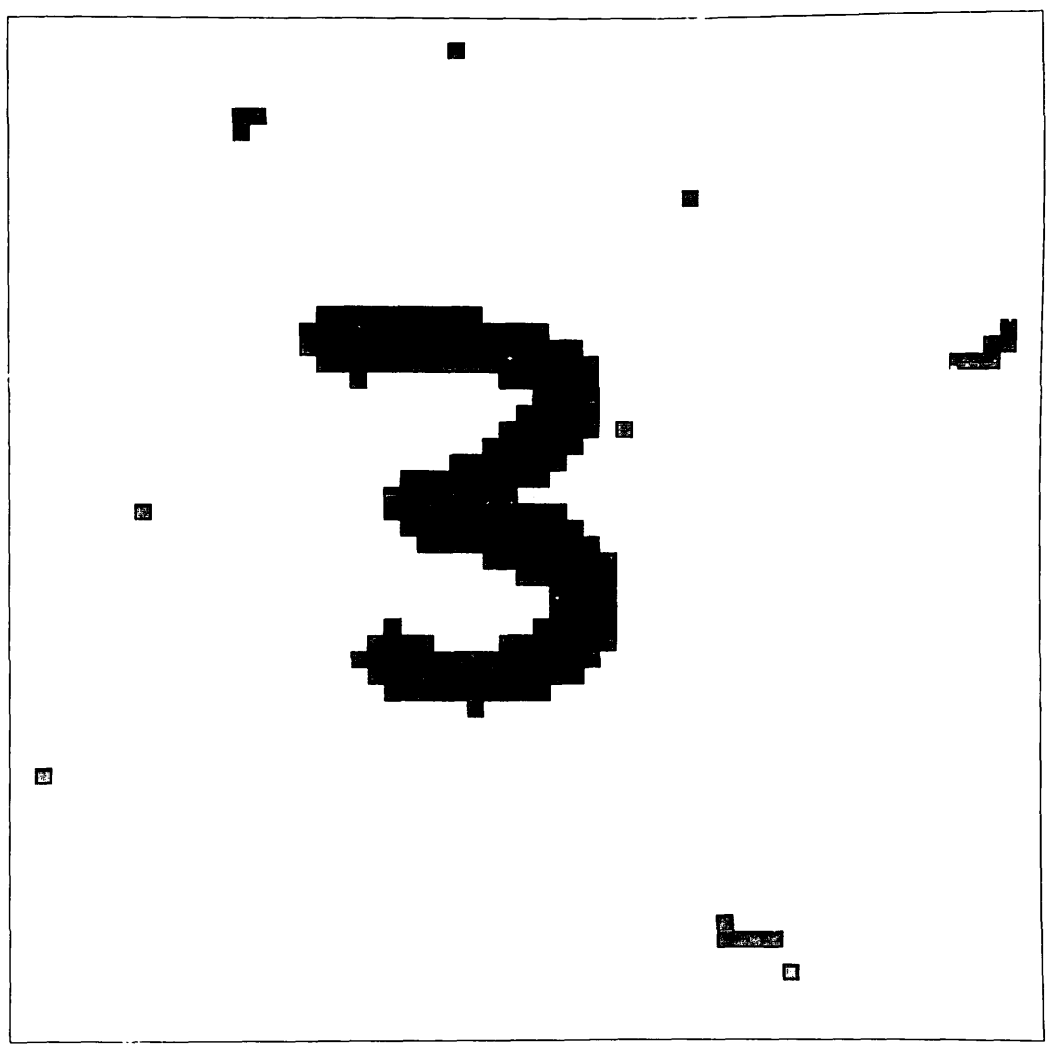


Fig. 4.2a Binary representation of a numeral '3' before noise reduction

B4086

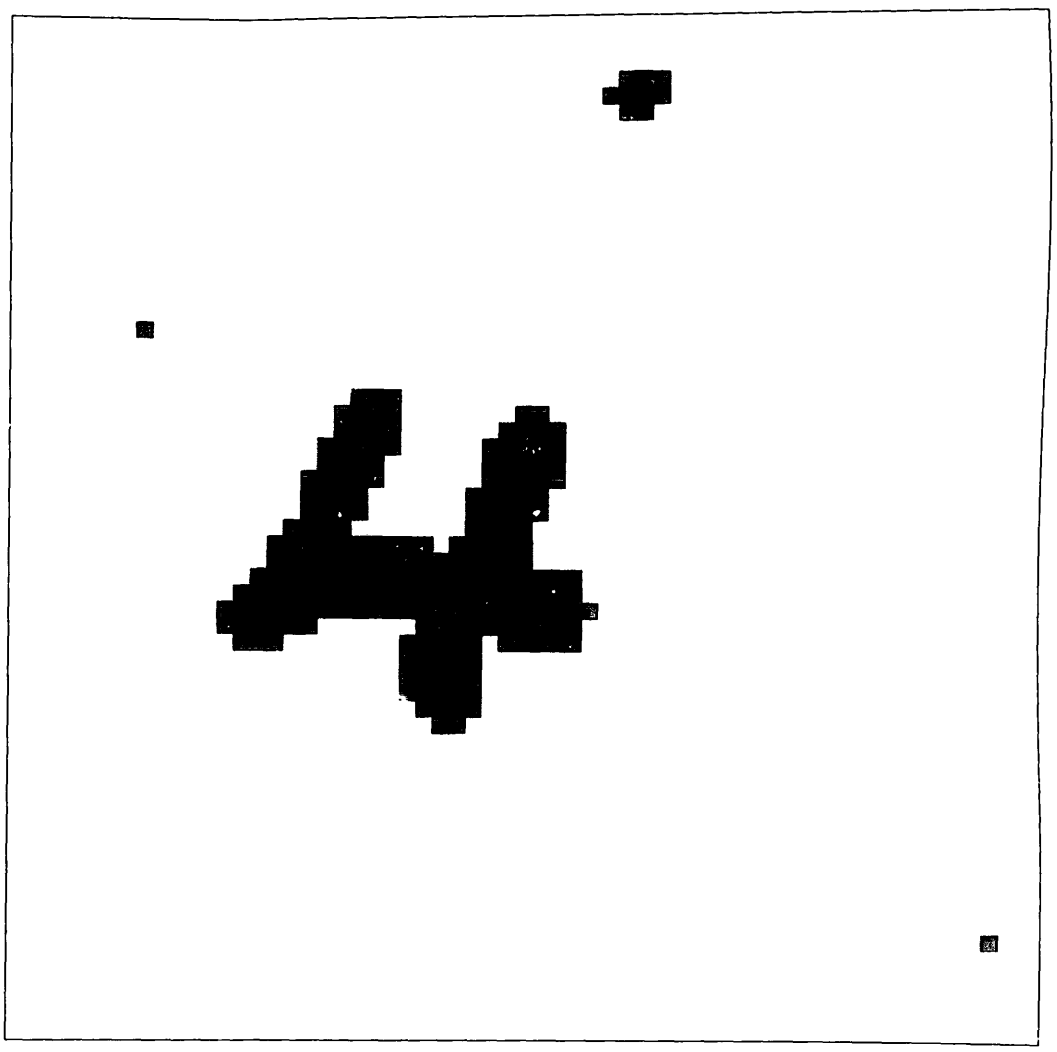


Fig. 4.2b Binary representation of a numeral '4'
before noise reduction

The Algorithm

Let $A = (a_{ij})$ denote the character matrix under consideration. In the present case A is of order 60 x 60. To remove the noise specks from A the following steps are followed:

Step 1 Augment the matrix by adding three rows in the end, two columns in the beginning and two columns in the end such that all the elements of these added rows and columns are 0's. Thus $A = (a_{ij})$ is now a matrix of order 63 x 64 with $i = 1,2,\ldots,63$ and $j = -1,0,1,\ldots,62$.

Step 2 Equate all the elements of the first row to 0, i.e., $a_{ij} = 0$, $j = -1,0,1,\ldots,62$.

Step 3 For each $i = 2,3,\ldots,60$ and $j = 1,2,\ldots,60$ check whether $a_{ij} = 0$ or 1. If $a_{ij} = 0$ then goto next element. If $a_{ij} = 1$ then check for conditions (1) to (14) given in Fig. 4.3. If any of these conditions is true then make $a_{ij} = 0$, otherwise no operation. Then go to next element.

The matrix augmentation made in step 1 is required for checking the conditions mentioned in step 3. In step 2 all the elements of the first row are made equal to 0. This is based on the assumption that while writing a character the author usually leaves some blank space on the top. In the case of doubt about the validity of this assumption it can be easily avoided by adding a '0'-row in the beginning of the matrix A. In step 3 the 14 conditions regarding the sum of boundary elements in windows of different sizes are checked.

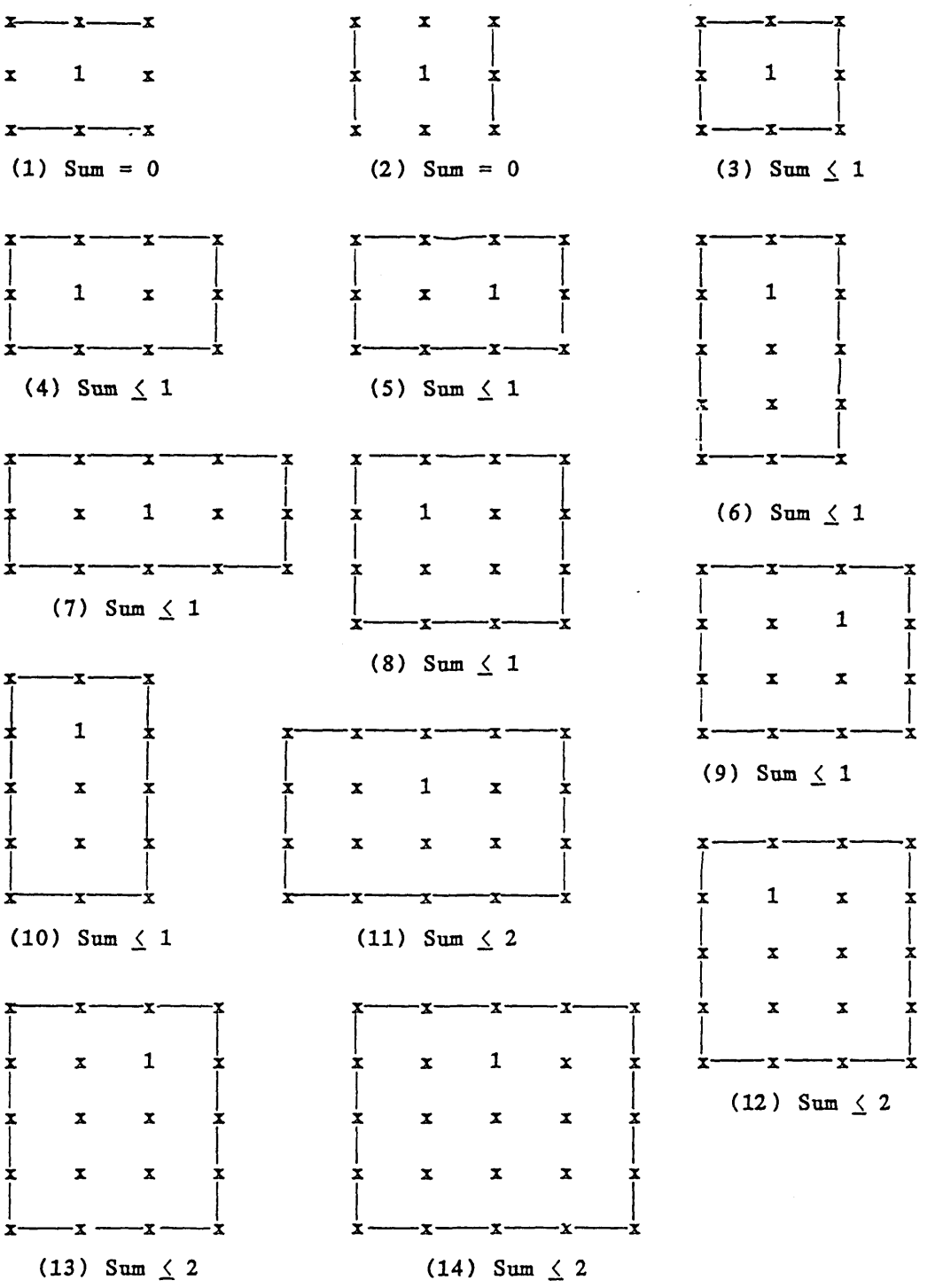


Fig. 4.3 Noise conditions for a black ('1') element (the term 'Sum' stands for the sum of connected elements).

Fig. 4.3 is more or less self-explanatory. The '1' inside a window represents the element under consideration. The boundary elements denoted by 'x' and connected by straight line segments are summed up to decide whether '1' forms a part of the character or it is a noise. Elements marked as 'x' in Fig. 4.3 could be either '0' or '1'. In condition (1) it is checked if all the six elements in the previous and the following rows are '0'. In condition (2) similar check is made for columns. In conditions (3) to (10) it is checked if at the most one of the surrounding 'x'-elements is '1'. In conditions (11) to (14) it is checked if the sum of the surrounding 'x'-elements is less than or equal to 2. If any of the above fourteen conditions holds then it is highly unlikely that the element under consideration constitutes a part of the character. In this case, therefore, it is replaced by '0'.

The program incorporating the above noise reduction algorithm is listed in Appendix B 3. Some idea about the effectiveness of the algorithm can be had from Figures 4.4a and 4.4b which show the representations of the numerals of Figures 4.2a and 4.2b after the application of the algorithm.

After the application of the noise reduction algorithm , the character samples were identified by C0011, C0012, ..., C0110; C1011, ..., C1110; C9011, ..., C9110.

C3045

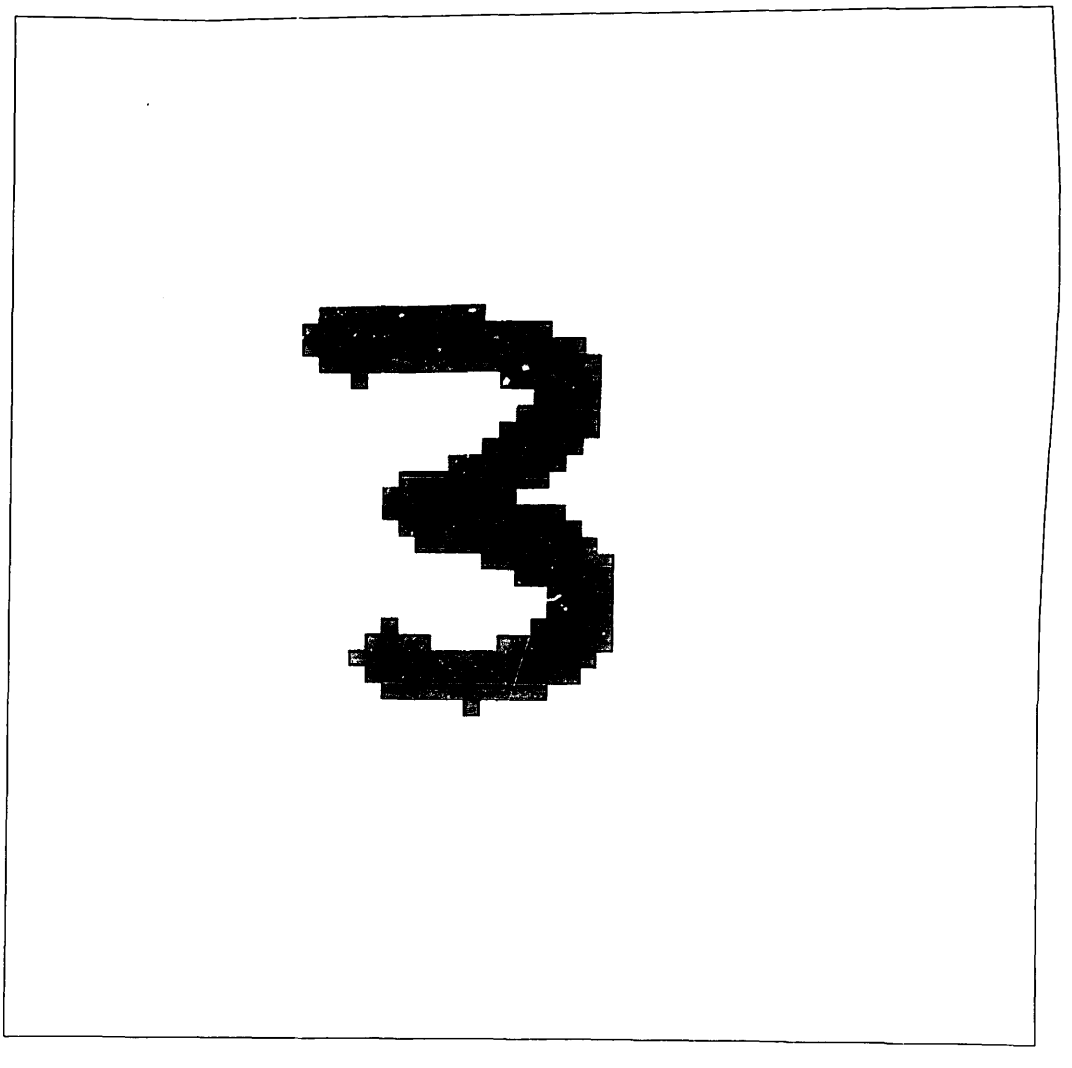


Fig. 4.4a Binary representation of the numeral '3' of Fig. 4.2a after noise reduction

C4086

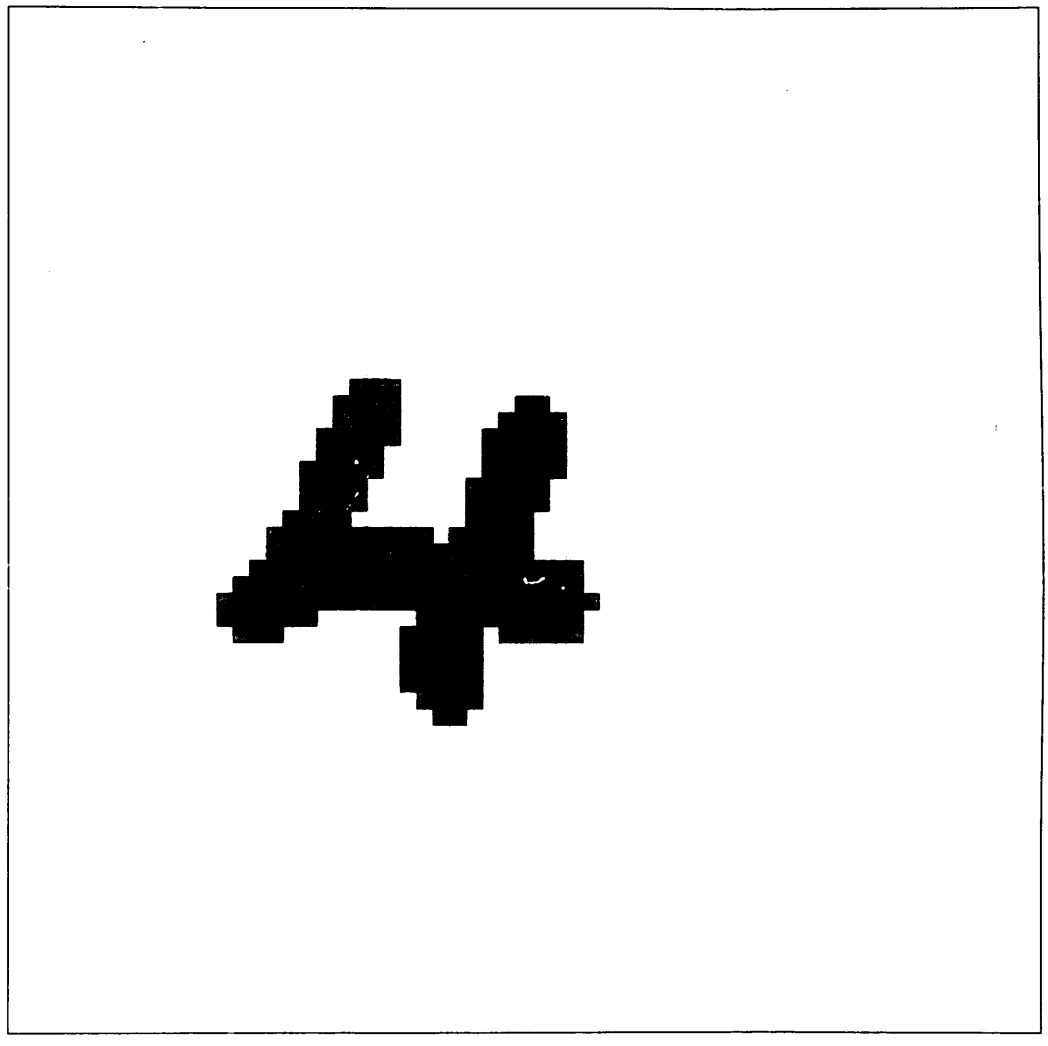


Fig. 4.4b Binary representation of the numeral '4' of Fig. 4.2b after noise reduction

4.4 Feature Extraction

Feature extraction poses the most serious problem in the design of a system intended to recognize handprinted characters. There are at least two reasons for this: (i) it is not clearly known what features are used by a human being who is capable of recognizing characters so accurately in spite of their variations in shape and size, and (ii) though it is believed that the curvature of line segments, the gap between lines, the slant, geometry and topology of the character, linguistic information, etc. form some useful cues, computerization of them is not easy. In an attempt to overcome the above difficulties, various features have been suggested by researches in the past. An excellent review of features used in the recognition of handprinted characters has been made by Suen [114] who has categorized the features into two broad groups, namely, global features and structural features. Global features are those which are extracted from every element which lies within a rectangle circumscribing the character. They do not reflect any local, geometrical or topological properties of the drawing itself [110]. Global features include n-tuples [6], [107], [115], [116], moments [117], [118], frequencies (i.e., numbers) of occurences of black elements in different regions of the rectangle containing a character [119], crossings and distances [5], characteristic loci [120] - [123] and transformations and series expansions [124] - [130]. Structural features include edges and line segments [131] - [133] and features generated by various contour tracing methods [134] - [140].

Since the structural features, also known as syntactic features [141], are derived from the topology of a handprinted character, they have gained more popularity than the global features. In their paper [142], which won the Seventh Annual Pattern Recognition Society Award, Duerr, etal. have proposed a hierarchically structured system for recognition of unconstrained handwritten numerals which consists of a conventional statistical classifier and a structural classifier analyzing the topological composition of patterns. In the present study, however, two sets of global features, designated as the 'normalized frequency' features and the 'normalized characteristic loci' features, were used for the recognition of handprinted numerals. The features chosen for study in this thesis were not intended mainly to form the basis of a practical scheme but more, as mentioned in the introductory section of the present chapter, they were chosen as an aid to the comparison of existing and proposed feature evaluation methods. The above mentioned two sets of features were well suited for the implementation of the feature evaluation methods under investigation, and, they were easy to extract. Nevertheless, experimental results obtained in the present study have shown that in character recognition these features are, however, quite effective. Development of the features will now be discussed.

4.4.1 'Normalized Frequency' Features

From each character matrix (of order 60 x 60) the rectangle circumscribing the numeral was determined by locating the minimum row number, the maximum row number, the minimum column number and the maximum column number having '1's present in them. This rectangle, which henceforth will be called the 'character rectangle', was then subdivided into 20 rectangular boxes by making 4 equal vertical divisions and 5 equal horizontal divisions. When the number of rows was not divisible by 5, extra rows containing '0's were added at the beginning and at the end of the character rectangle and the next higher multiple of 5 was considered as the number of rows. Similar modifications were made for columns. The frequency of '1's present in each of these 20 boxes was then counted. Several authors [43], [143] have used similar frequency counts as features. But they dealt with a fixed size of the rectangle for all the characters under consideration. To reduce the effect of the size of a character on its feature values each of the 20 frequencies was divided by the total (of the 20 frequencies). The resulting 20 proportions constituted, what was called, the normalized frequency features.

The program for extraction of the normalized frquency features is listed in Appendix B 4.

4.4.2 'Normalized Characteristic Loci' Features

The 'characteristic loci' features were devised by Glucksman [120] for the recognition of machine printed characters. Knoll [122], Michael and Lin [123], Spanjersberg [117], and Kwan et al.[144] applied them for the recognition of handprinted characters.

'Characteristic loci' are a set of 4-digit codes generated from the white points ('0'-valued pixels) contained in the character rectangle. For a white point each of these four digits contains the number of line crossings from the point to one of the four perpendicular directions: left, top, right and bottom. The count is restricted to 2 for two or more crossings in any one direction. The four digits form a ternary code. For example, in Fig. 4.5 the white point A has ternary code 0112 while the point B gives rise to the code 1101. The total number of points in the character rectangle matching a given code determines the value of the feature corresponding to that code. Since a four digit ternary code can have values in the range 0 to 80, the above feature generation procedure leads to a maximum of 81 'characteristic loci' features.

The sum of the above 81 features will be equal to the number of '0'-pixels in a character rectangle. To reduce the effect of the size of a character on its feature values each characteristic loci feature was divided by the above sum. The resulting proportions constituted, what was called, the 'normalized characteristic loci' features.

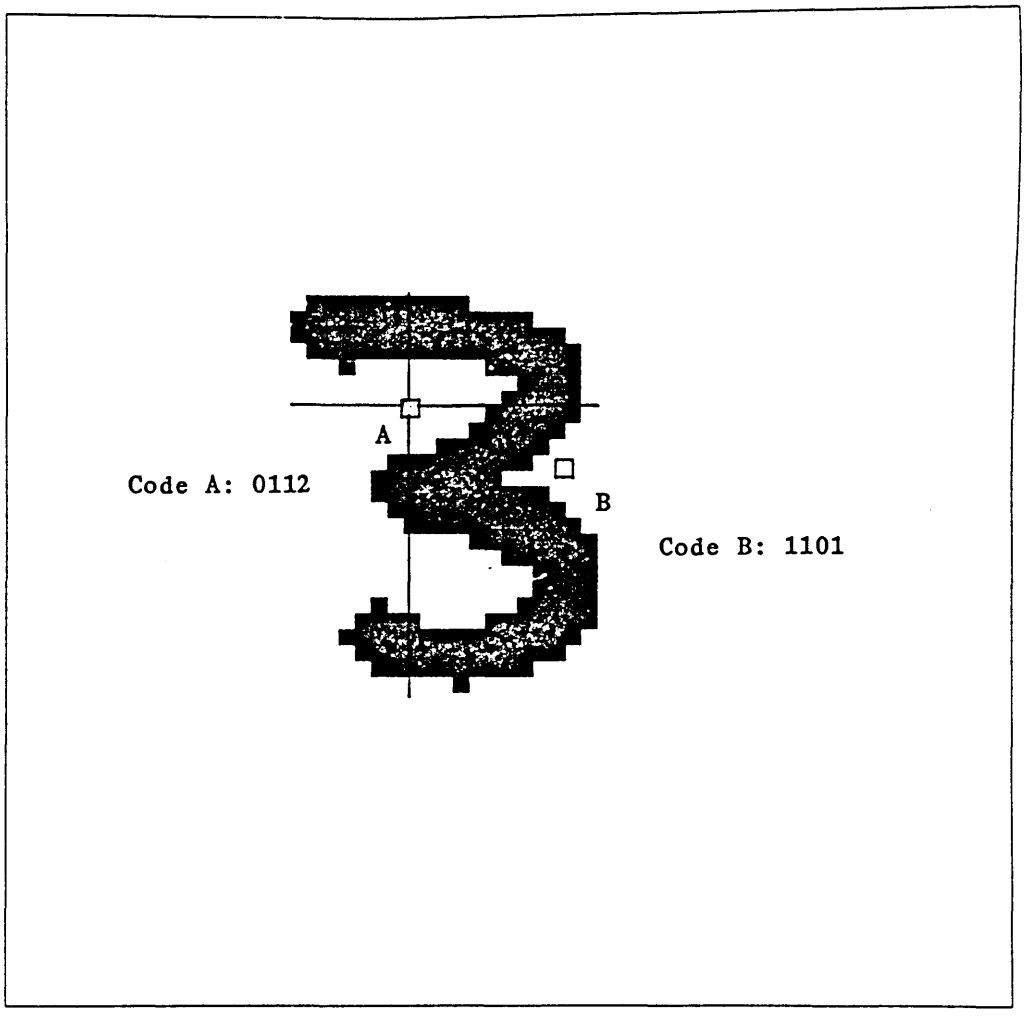


Fig. 4.5 Characteristic loci codes for points A and B in a sample character '3'

The program for the development of the characteristic loci features is given in Appendix B 5.

4.5 Deletion of Redundant Features

Initially 20 normalized frequency features and 81 normalized characteristic loci features , that is, a total of 101 features were considered. Some of these features would be redundant because, for example, as the characteristic loci features were calculated only on the basis of white points lying within the character rectangle, the codes with three or four '0's in them could not occur at all. To mention more explicitly, the nine codes 0000, 0001, 00002, 0010, 0020, 0100, 0200, 1000 and 2000 could not occur for any character.

For the detection of redundant features a simple procedure was followed: for each of the 101 features all its values were added over the whole data set of 1000 characters and the features with zero sums were decided to be redundant.

As can be seen from Table 4.1, 23 characteristic loci features, including the nine mentioned above, turned out to be redundant. It may also be noted from Table 4.1 that another 25 features made very little contribution to the grand total of 1000.000 in the sense that each of them had a sum value of less than 1.000. Though this raised some doubt about their usefulness, at this stage there was no reason to consider them to be redundant.

Table 4.1
Feature sum over 1000 character samples for each of the 81 characteristic loci codes

Serial No.	Code	Sum	Serial No.	Code	Sum	Serial No.	Code	Sum
1	0000	0	28	1000	0	55	2000	0
2	0001	0	29	1001	73.026	56	2001	19.316
3	0002	0	30	1002	17.958	57	2002	1.049
4	0010	0	31	1010	0	58	2010	0
5	0011	114.389	32	1011	2.911	59	2011	0.164
6	0012	48.479	33	1012	1.200	60	2012	0.023
7	0020	0	34	1020	0	61	2020	0
8	0021	26.201	35	1021	0.086	62	2021	0
9	0022	5.049	36	1022	0.016	63	2022	0
10	0100	0	37	1100	149.591	64	2100	28.046
11	0101	0	38	1101	42.923	65	2101	5.277
12	0102	0	39	1102	23.411	66	2102	0.430
13	0110	75.074	40	1110	1.324	67	2110	0.142
14	0111	55.549	41	1111	81.118	68	2111	2.471
15	0112	17.083	42	1112	27.247	69	2112	0.395
16	0120	6.270	43	1120	0.195	70	2120	0
17	0121	2.831	44	1121	1.476	71	2121	0.009
18	0122	0.370	45	1122	0.682	72	2122	0
19	0200	0	46	1200	63.683	73	2200	5.846
20	0201	0	47	1201	6.559	74	2201	0.424
21	0202	0	48	1202	1.084	75	2202	0.048
22	0210	28.232	49	1210	1.936	76	2210	0.063
23	0211	29.110	50	1211	28.863	77	2211	0.590
24	0212	0.546	51	1212	0.187	78	2212	0.004
25	0220	0.448	52	1220	0.100	79	2220	0
26	0221	0.298	53	1221	0.186	80	2221	0.004
27	0222	0.008	54	1222	0.005	81	2222	0
						Total		1000.000

Since none of the 20 normalized frequency features gave rise to a zero sum, a total of 78 (= 101 - 23) features were considered for further analysis discussed in chapters 5 and 6.

CHAPTER 5

FEATURE ORDERING EXPERIMENTS

5.1 Introduction

This chapter is concerned with feature orderings obtained by the employment of the two-class and the multiclass probabilistic criteria discussed in chapter 2 and the Mahalanobis distance-based criteria discussed in chapter 3.

Section 5.2 deals with the implementation of probabilistic feature evaluation criteria. Estimation of class-conditional probability distributions, a prerequisite for their implementation, is discussed in subsection 5.2.1. Feature ordering experiments are then conducted in subsection 5.2.2.

Section 5.3 deals with the implementation of Mahalanobis distance-based criteria. Estimation of means and covariances, required for the implementation of the criteria, is dealt with in subsection 5.3.1. Subsection 5.3.2 deals with the feature ordering experiments based on Mahalanobis distance-based criteria.

In section 5.4 the feature orderings are compared by analyzing the rank correlation coefficients between various pairs of orderings. The rank correlation coefficients are tested for their statistical significance. Significance of a rank correlation coefficient between two orderings would indicate their conformity with each other. To get a clearer picture of the extent of conformity between orderings, in addition to the orderings obtained by various feature evaluation

criteria, a random ordering of the features is also included in the rank correlation analysis.

The features dealt with were the normalized frequency features and the normalized characteristic loci features extracted from the data on ten numeric characters 0, 1, ..., 9. To take into account the style variations in writing the numerals 1, 4 and 7, each of them was treated to be composed of two different scripts. Considering each script as a separate class, therefore, there were a total of 13 classes. Table 5.1 shows the frequency distribution of the data set of 1000 numeral samples in these 13 classes. Both the classes 2 and 3 represented the same numeral '1'. Similar was the case with the class-pairs (6,7) and (10,11) which represented the numerals '4' and '7', respectively. In employing a two-class feature evaluation criterion for measuring the effectiveness of features in the 13-class numeral recognition problem the contributions of the above three class-pairs were ignored. By this the contributions coming out of the discrimination between classes representing the same numeral could be avoided.

5.2 Feature Orderings by Probabilistic Criteria

5.2.1 Estimation of Class-conditional Probability Distributions

In view of the limited size of the data set the features were assumed to be independent. This was necessary to avoid inaccuracies involved in a small sample estimation problem. Under the assumption of independence of features the number of class-conditional probability distributions required to be estimated was 78 x 13 = 1014.

Table 5.1 Frequency distribution of 1000 character samples in 13 script classes

Numeral	Script Class		Character identifiers	No. of obs.
(1)	(2)	(3)	(4)	(5)
0	0	1	C0011, C0012, ..., C0110	100
1	1	2	C1011, ..., C1020, C1031, ..., C1040, C1051, ..., C1110	80
	1	3	C1021, ..., C1030, C1041, ..., C1050	20
2	2	4	C2011, C2012, ..., C2110	100
3	3	5	C3011, C3012, ..., C3110	100
4	4	6	C4011, ..., C4040, C4055, C4060, C4071, ..., C4110	72
	4	7	C4041, ..., C4054, C4056, ..., C4059, C4061, ..., C4070	28
5	5	8	C5011, C5012, ..., C5110	100
6	6	9	C6011, C6012, ..., C6110	100
7	7	10	C7011, ..., C7020, C7041, ..., C7090	60
	7	11	C7021, ..., C7040, C7091, ..., C7110	40
8	8	12	C8011, C8012, ..., C8110	100
9	9	13	C9011, C9012, ..., C9110	100

In general, for an m-class problem with n features, the number of distributions is n.m.

A histogram approach with equal intervals [91] was used for estimation of probability distributions. Computational simplicity and distribution-free nature of the approach were the justifications for its use. The range of values of a feature (over all classes) was divided into a number of mutually exclusive intervals of equal length. For each class the proportion of observations lying in each of these intervals was then obtained. These proportions gave the estimates of proabilities corresponding to the above intervals. The intervals were represented by their mid-points. Effectively, a 'discrete' approximation of a probability distribution of 'continuous' type was obtained. The procedure was repeated for all the 78 features.

A computer program for the estimation procedure described above is listed in Appendix B 6. The program is designed to work for one set of observations, that is, observations belonging to one class. To obtain the estimates for different classes the program was executed repeatedly. It is easy to generalize the program to work for any number of classes. The number of features, the number of observations on the basis of which the estimation is to be made, and the number of intervals are supplied interactively. In the experiments under consideration the number of intervals was taken to be equal to 10. Minimum and maximum values of each feature obtained from the whole data set of 1000 observations were used in the program as the range of values of the feature.

A drawback of the histogram approach with equal intervals is that it requires too much storage. Even in the case of independent features an array of size m.n.v is needed to store probability estimates in an m-class problem with n features and v intervals. In absence of the assumption of independence of features the storage requirement increases rapidly to $m.v^n$. In this case, therefore, it is not advisable to use this approach. Further details about the approaches of estimation of probability distributions are beyond the scope of the present study. Some of the suggested readings in this area are [91], [100] and [145] - [150].

5.2.2 Implementation of Probabilistic Criteria

Experimental investigation included the following 2-class and m-class criteria:

2-class criteria: the Bayesian probability of error (2-class); the Bhattacharyya coefficient; Jeffreys-Matusita distance function and the divergence function.

m-class criteria: the Bayesian probability of error (m-class); Matusita's measure of affinity; Shannon's conditional entropy and the Bayesian distance of Devijver.

Application of a 2-class criterion in an m-class problem is not straightforward and there are at least two approaches for this purpose. These two approaches are commonly known as the maximum expected value approach and the maximin approach. With m classes there can be $^mC_2$ class-pairs. In the expected value approach, for each feature set to be evaluated, the average of all the $^mC_2$ values of

the 2-class feature evaluation criterion under consideration are computed, and the feature set with the maximum average value is considered to be the best feature set. In the maximin approach, for each feature set, the minimum of the ${}^{m}C_{2}$ criterion values is obtained. Then the feature set with the maximum of the above minimum values is considered to be the best feature set.

The above two approaches apply if the criterion under consideration is such that higher criterion value corresponds to more class separa tion. If higher criterion value corresponds to less class separation then the approaches followed are the minimum expected value approach and the minimax approach.

In the present experiments with 2-class criteria the expected value approach was used. This approach is found to be more popular [71] of the two approaches described above. For reasons mentioned earlier, the three class-pairs (2,3), (6,7) and (10,11) were excluded from the computation of the average value of a criterion. The average value was computed based on the remaining ${}^{13}C_{2} - 3$, that is, 75 class-pairs.

Criterion values for class-pairs (C_i, C_j) were computed assuming the a'priori probabilities of the classes to be the same. For a feature, say X, the 2-class probabilistic criteria, defined in continuous form in chapter 2, were computed using the following discrete approximations:

Bayesian probability of error (2-class)

$$P_{eij} = \frac{1}{2} \sum_{u=1}^{v} \min_{i,j} \{ p(x_u|C_i), p(x_u|C_j) \} \qquad (5.1)$$

Bhattacharyya coefficient

$$\rho_{ij} = \sum_{u=1}^{v} [p(x_u|C_i) \cdot p(x_u|C_j)]^{1/2} \qquad (5.2)$$

Jeffreys-Matusita distance function

$$\gamma_{ij} = \sqrt{2 (1 - \rho_{ij})} \qquad (5.3)$$

Divergence function

$$J_{ij} = \sum_{u=1}^{v} [p(x_u|C_i) - p(x_u|C_j)] \ln \left[\frac{p(x_u|C_i)}{p(x_u|C_j)} \right] \qquad (5.4)$$

In the above equations v is the number of intervals in which the range of values of X is divided, x_u is the representative of the uth interval, and $p(x_u|C_i)$ is the (estimate of the) class-conditional probability of X corresponding to the uth interval. It may be noted here that the equations (5.1), (5.2) and (5.4) are obtained by replacing the integrations with summations and by putting $\pi_i = \pi_j = 1/2$ (applicable to 5.1 only) in the equations (2.2c), (2.4)

and (2.14), respectively.

The program for ordering the features in decreasing order of effectiveness using the above criteria is listed in Appendix B 7. This program also includes the ordering by the Kolmogorov variational distance. However, results obtained by the application of this criterion will not be reported here because, due to its exact linear relationship with the Bayesian error criterion, the feature ordering will be the same as that by the Bayesian error criterion.

Equations used for the discrete approximations of the m-class measures are given below:

Bayesian probability of error (m-class)

$$P_e = 1 - \sum_{u=1}^{v} \max \left[\pi_1 p(x_u|C_1), \ \pi_2 p(x_u|C_2), \ \ldots, \ \pi_m p(x_u|C_m) \right] \quad (5.5)$$

Matusita's measure of affinity

$$\rho_m^* = \left[\pi_1 \pi_2 \ldots \pi_m \right]^{\frac{1}{m}} \sum_{u=1}^{v} \left[p(x_u|C_1).p(x_u|C_2)\ldots p(x_u|C_m) \right]^{\frac{1}{m}} \quad (5.6)$$

Shannon's conditional entropy

$$H = \sum_{u=1}^{v} \left\{ \left[\sum_{i=1}^{m} \pi_i \, p(x_u|C_i) \right] \cdot \ln \left[\sum_{i=1}^{m} \pi_i \, p(x_u|C_i) \right] - \sum_{i=1}^{m} \left[\pi_i \, p(x_u|C_i) \right] \cdot \ln \left[\pi_i p(x_u|C_i) \right] \right\} \quad (5.7)$$

Devijver's Bayesian distance

$$B = \sum_{u=1}^{v} \left\{ \left(\sum_{i=1}^{m} \left[\pi_i \, p(x_u|C_i) \right]^2 \right) \Big/ \left(\sum_{i=1}^{m} \left[\pi_i p(x_u|C_i) \right] \right) \right\} \quad (5.8)$$

The program for arranging the features in decreasing order of effectiveness using the equations (5.5) to (5.8) is listed in the Appendix B 8. The program works for any values of the a'priori probabilities. In the present experiments they were assumed to be equal. In addition to giving the orderings of the features the program also gives the values of the criteria for different features.

Table 5.2 shows the feature orderings obtained by various 2-class and m-class probabilistic criteria. To indicate the adoption of the expected value approach, a bar ('-') is put on the symbols denoting the 2-class criteria. Further analysis of these orderings will be made in sction 5.4, and also in the next chapter.

Table 5.2
Feature orderings (in descending order of effectiveness) by probabilistic criteria

Rank	$\bar{P}_{e2}$	$\bar{\rho}$	$\bar{\gamma}$	$\bar{J}$	P_e	ρ_m^*	H	B
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
1	13	13	13	13	18	47	13	43
2	17	17	17	17	43	43	43	47
3	2	26	2	26	11	20	17	18
4	43	1	11	2	13	18	18	1
5	26	2	18	1	9	17	1	13
6	11	18	43	43	2	13	2	31
7	18	11	26	18	3	12	26	25
8	9	43	1	11	10	11	20	20
9	12	20	9	9	4	9	11	2
10	10	9	21	20	1	8	25	11
11	21	21	10	8	7	1	9	17
12	1	12	20	12	21	5	31	9
13	16	8	12	10	12	2	21	37
14	25	25	8	25	8	7	10	21
15	8	10	23	21	25	10	12	26
16	14	23	16	31	6	22	8	10
17	20	31	25	23	19	6	16	3
18	23	16	14	16	16	14	23	16
19	22	7	7	47	17	26	3	7
20	19	14	3	14	22	21	7	8
21	7	3	31	7	20	15	14	45
22	6	61	6	3	31	16	47	12
23	3	22	5	61	52	19	61	6
24	4	47	19	22	14	25	22	32
25	52	5	22	5	26	4	6	61
26	31	6	52	6	5	3	19	14
27	5	19	4	44	37	23	37	22
28	44	44	61	19	47	61	52	4
29	15	52	44	52	23	31	4	27
30	37	4	15	4	15	52	44	56
31	61	65	37	32	65	44	5	23
32	65	37	65	65	61	37	32	52
33	47	56	47	37	44	65	65	19
34	48	32	48	56	27	48	56	44
35	56	27	56	27	32	27	27	65
36	27	48	27	15	45	56	45	5
37	32	15	32	48	56	32	48	48
38	45	45	29	45	48	45	15	15
39	29	29	66	29	28	28	29	55
40	55	55	72	55	66	29	55	39
41	66	66	55	66	55	72	66	53
42	72	72	53	72	29	66	53	29
43	28	53	45	53	72	53	72	66
44	53	28	28	28	53	55	28	69
45	46	46	46	46	39	38	39	72
46	24	24	24	24	62	46	46	28
47	62	39	62	62	24	24	24	24
48	38	62	38	38	46	62	69	46
49	39	38	40	39	69	39	62	62
50	30	69	39	69	38	40	30	68
51	69	30	67	30	30	30	38	30
52	40	40	30	40	40	69	67	73
53	67	67	34	50	67	67	68	67
54	50	54	50	67	68	50	40	54
55	54	50	69	54	50	34	54	49
56	34	68	54	68	73	54	50	38
57	68	73	33	73	54	68	73	58
58	73	34	57	34	59	73	51	51
59	51	33	73	51	33	33	49	40
60	33	61	59	33	51	51	33	50
61	49	49	68	70	58	49	58	70
62	70	70	49	49	63	70	34	33
63	63	58	63	58	34	63	70	63
64	58	63	70	63	70	59	63	41
65	59	57	35	59	59	57	59	59
66	57	59	51	76	57	58	57	75
67	76	76	58	57	76	35	76	76
68	35	75	75	75	75	76	75	57
69	75	35	76	35	41	75	41	34
70	41	41	41	41	35	41	35	74
71	74	74	64	74	74	74	74	35
72	64	64	74	71	64	64	64	64
73	42	42	42	64	71	42	71	71
74	71	71	71	42	42	71	42	42
75	36	36	36	36	36	36	36	36
76	77	77	77	77	77	77	77	77
77	60	60	60	60	60	60	60	60
78	78	78	78	78	78	78	78	78

5.3 Feature Orderings by Mahalanobis Distance-Based Criteria

5.3.1 Estimation of Means and Covariances

Estimation of these parameters is required for the implementation of the Mahalanobis distance-based criteria. Unbiassed estimates of the mean vector and the dispersion matrix of the feature vector X in class i (i = 1,2,...,m) are given by

$$\bar{x}_i = \frac{1}{N_i} \sum_{k=1}^{N_i} x_{ik} \tag{5.9}$$

and

$$S_i = \frac{1}{N_i - 1} \sum_{k=1}^{N_i} (x_{ik} - \bar{x}_i)(x_{ik} - \bar{x}_i)' \tag{5.10}$$

where $x_{i1}, x_{i2}, \ldots, x_{iN_i}$ are the N_i sample observations (observation vectors) from the ith class. The n diagonal elements of S_i represent the estimates of the variances of the n features in the ith class. In the case of independence of features the above estimation problem reduces to the problem of estimation of means and standard deviations of the features in different classes.

5.3.2 Implementation of Mahalanobis Distance-Based Criteria

Implementation of the Mahalanobis distance-based criterion D_A^2, defined in chapter 3 (Equation 3.21), will be described in details. Implementation of D_B^2 (Equation 3.22) and D^2 (Equation 3.9) will

require similar steps.

For a class-pair (C_i, C_j), with equal a'priori probabilities, the D_A^2 criterion is given by

$$D_{Aij}^2 = \frac{D_{ij}^2}{4 + D_{ij}^2} \tag{5.11}$$

As with the probabilistic criteria, the expected value approach described in the previous section are used to obtain a combined criterion value. Thus, D_A^2 is obtained by using the following equation:

$$D_A^2 = \frac{1}{{}^mC_2} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} D_{Aij}^2 \tag{5.12}$$

In the present experiments the features were ordered in two stages. In the first stage all the 78 features were arranged assuming them to be independent. In the second stage the first 25 features were rearranged by taking into account their covariance terms and following a step-by-step procedure of inclusion of features starting with one feature. The decision to deal with 25 features in the second stage was made on the basis of computer memory and computational requirements. The following is a brief account of the tasks involved in the above mentioned stages for the implementation of the D_A^2 criterion.

Stage I

The features are assumed to be independent and they are dealt with individually. Let X be a feature to be evaluated. It may be noted here that for notational simplicity the same symbol X is used to represent a feature which was earlier used to represent a feature vector. The Mahalanobis distance for a class-pair (C_i, C_j) is then obtained by using

$$D_{ij}^2 = \frac{(\bar{x}_i - \bar{x}_j)^2}{s_{ij}^2} \tag{5.13}$$

where $\bar{x}_i$ and $\bar{x}_j$ are the sample means of the feature X in the two classes and s_{ij}^2 is the average of the two within-class sample variances. s_{ij}^2 is given by

$$s_{ij}^2 = \frac{1}{N_i + N_j - 2} \left[(N_i - 1)\, s_i^2 + (N_j - 1)\, s_j^2 \right] \tag{5.14}$$

where

$$s_i^2 = \frac{1}{N_i - 1} \sum_{k=1}^{N_i} (x_{ik} - \bar{x}_i)^2 \tag{5.15}$$

and

$$s_j^2 = \frac{1}{N_j - 1} \sum_{k=1}^{N_j} (x_{jk} - \bar{x}_j)^2 \tag{5.16}$$

The above value of D^2_{ij} is put in the equation (5.11) to get the value of D^2_{Aij}. The values of D^2_{Aij} are then put in the equation (5.12) to get the combined criterion value D^2_A.

As in the case of probabilistic measures, in the actual experiments 3 class-pairs were excluded and the above combined criterion value was obtained on the basis of the remaining 75 class-pairs. The above process was repeated to obtain the values of the D^2_A criterion for all the 78 features. The features were then arranged in decreasing order of their D^2_A values.

The program for the implementation of the above procedure is listed in Appendix B 9. Feature orderings obtained are given in column (2) of Table 5.3. The values of the D^2_A criterion for the features are shown in column (3) of the table in order to give a comparative picture of their effectiveness. As can be seen from the table, the top ranking feature (feature no. 13) has a criterion value of 0.363. The minimum possible value of 0 is taken by the features 36,77 and 78, showing their total uselessness for the purpose of numeral classification.

Orderings by D^2_B criterion and D^2 criterion, also with the assumption of indepenence of features, are shown in the same table. The orderings by D^2_A, D^2_B and D^2 are denoted by $D^2_A(1)$, $D^2_B(1)$ and $D^2(1)$, respectively. This is to distinguish them from further orderings to be made in stage II wherein the features are no more assumed to be independent.

Table 5.3
Feature orderings by Mahalanobis distance-based criteria under the assumption of independence of features

Rank	D_A^2 criterion		D_B^2 criterion		D^2 criterion		A random ordering
	Ordering ($D_A^2(1)$)	Criterion value	Ordering ($D_B^2(1)$)	Criterion value	Ordering ($D^2(1)$)	Criterion value	(R)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	13	.363	13	.317	13	6.91	46
2	26	.353	26	.303	20	6.37	53
3	2	.322	1	.290	1	5.70	73
4	17	.320	2	.268	18	5.08	3
5	1	.319	17	.267	47	4.97	32
6	9	.317	9	.262	11	4.59	4
7	18	.307	20	.256	26	4.59	57
8	43	.304	18	.255	9	4.09	14
9	44	.298	43	.246	17	4.09	68
10	20	.296	44	.242	12	3.87	59
11	11	.290	47	.241	8	3.41	10
12	21	.286	11	.240	2	3.33	63
13	8	.286	31	.236	43	2.89	78
14	12	.281	8	.235	31	2.87	56
15	23	.273	12	.226	44	2.87	58
16	31	.272	21	.224	21	2.46	27
17	47	.270	25	.215	25	2.45	16
18	25	.267	14	.210	14	2.39	29
19	14	.266	23	.204	10	2.36	52
20	16	.258	10	.203	61	2.17	1
21	10	.255	61	.199	23	2.17	74
22	61	.255	16	.196	22	2.10	17
23	32	.250	22	.183	16	2.09	30
24	22	.244	32	.183	3	1.92	76
25	52	.219	3	.169	32	1.80	75
26	3	.219	52	.155	5	1.64	70
27	37	.209	37	.150	52	1.53	6
28	56	.209	5	.146	37	1.47	9
29	5	.198	56	.137	7	1.38	18
30	6	.193	7	.136	6	1.31	67
31	65	.189	6	.135	4	1.29	61
32	7	.188	65	.132	65	1.28	31
33	4	.188	4	.132	56	1.25	44
34	27	.187	27	.130	27	1.25	77
35	19	.177	45	.118	45	1.13	45
36	48	.173	19	.116	19	1.07	5
37	45	.167	48	.108	48	.959	50
38	55	.143	55	.0897	55	.796	20
39	66	.138	66	.0857	66	.759	11
40	15	.121	15	.0770	15	.697	23
41	72	.104	72	.0606	72	.518	38
42	29	.101	29	.0585	29	.495	25
43	53	.0883	53	.0511	53	.434	64
44	38	.0837	38	.0499	38	.431	72
45	28	.0738	28	.0421	28	.355	71
46	46	.0731	46	.0407	46	.339	34
47	24	.0691	24	.0377	24	.312	69
48	69	.0634	39	.0358	39	.308	65
49	62	.0628	62	.0353	62	.296	22
50	39	.0605	69	.0353	69	.294	15
51	50	.0440	50	.0234	34	.195	13
52	34	.0408	34	.0231	50	.191	2
53	40	.0376	30	.0200	30	.167	41
54	30	.0364	40	.0198	40	.161	62
55	51	.0328	51	.0172	51	.140	47
56	68	.0306	68	.0163	68	.134	24
57	35	.0304	35	.0161	35	.131	42
58	67	.0277	67	.0145	67	.118	39
59	49	.0246	49	.0130	49	.106	55
60	54	.0238	54	.0124	54	.101	21
61	70	.0224	58	.0117	58	.0953	54
62	58	.0224	70	.0116	70	.0944	8
63	73	.0222	73	.0115	73	.0935	60
64	75	.0196	75	.0102	75	.0830	12
65	63	.0175	63	.00912	63	.0739	35
66	33	.0172	33	.00880	33	.0710	26
67	76	.0148	76	.00756	76	.0610	33
68	59	.0120	59	.00614	59	.0495	36
69	57	.0112	57	.00571	57	.0459	7
70	41	.00933	41	.00482	41	.0390	66
71	74	.00533	74	.00273	74	.0221	40
72	60	.00523	60	.00268	60	.0216	43
73	42	.00454	42	.00230	42	.0185	51
74	64	.00444	64	.00225	64	.0181	19
75	71	.00375	71	.00191	71	.0154	49
76	78	0	78	0	78	0	28
77	77	0	77	0	77	0	48
78	36	0	36	0	36	0	37

Further analysis of the above orderings is postponed till the next section.

Stage II

Unlike in stage I, in this stage the covariances between the features are taken into account for the computation of the Mahalanobis distance-based criteria. For a feature vector X the sample Mahalanobis distance D^2 is given by equation (3.9). For a class-pair (C_i, C_j) the distance function can be written as

$$D_{ij}^2 = (\bar{x}_i - \bar{x}_j)' \; S_{ij}^{-1} \; (\bar{x}_i - \bar{x}_j) \qquad (5.17)$$

where the notations have the meanings similar to those in equations (3.10) to (3.14).

It may be worthwhile to mention here that in actual computations S_{ij}^{-1} is replaced by S_{ij}^{-}, the generalized inverse of S_{ij} [151],[152]. By this the computational problems arising out of attempts to compute the inverse of a singular matrix could be avoided.

To select a feature subset of size r, say, from a set of n features using the D_A^2 criterion the following step-by-step procedure is followed:

Step 1 Evaluate all the n features $X_1, X_2, \ldots, X_n$ using the criterion D_A^2 (averaged over all the class-pairs as in stage I). Select the feature with the maximum D_A^2 value. Suppose X_{k_1} gets selected by this process.

Step 2 Evaluate all the n-1 feature subsets of size 2 whose first feature is X_{k_1}. They are (X_{k_1}, X_1), (X_{k_1}, X_2), ..., (X_{k_1}, X_{k_1-1}), (X_{k_1}, X_{k_1+1}), ..., (X_{k_1}, X_n). Select the feature subset with the maximum D_A^2 value. Suppose it is (X_{k_1}, X_{k_2}).

.

.

.

Step r Evaluate all the (n-r+1) feature subsets of size r whose first r-1 features are X_{k_1}, X_{k_2}, ..., $X_{k_{r-1}}$. Select the feature subset $(X_{k_1}, X_{k_2}, \ldots, X_{k_{r-1}}, X_{k_r})$ having the maximum D_A^2-value.

The program for the above procedure is listed in Appendix B 10. The program uses the D_A^2 criterion. It can easily be modified to work for any other D^2-based criterion. Input and output files used and/or created in the program are shown in Fig. 5.1. The program selects feature subsets of all the sizes 1,2,...,n.

In the present experiments the D^2-based criteria were implemented in stage II using the 25 features obtained by D_A^2 criterion in stage I as the initial set. Then, by taking n = 25, feature subsets of sizes 1, 2, , ..., 25 were selected. Tables 5.4a and 5.4b illustrate the results obtained by using the D_A^2 criterion and the D_B^2 criterion, respectively. The feature subsets and the corresponding criterion values are both shown in the tables.

The increases in values of D_A^2 and D_B^2 with increase in number of features are illustrated in Fig. 5.2. It will be noted from this diagram that the increases in values of D_A^2 and D_B^2, in relation to each other, are in conformity with those shown in Fig. 3.3 of chapter 3.

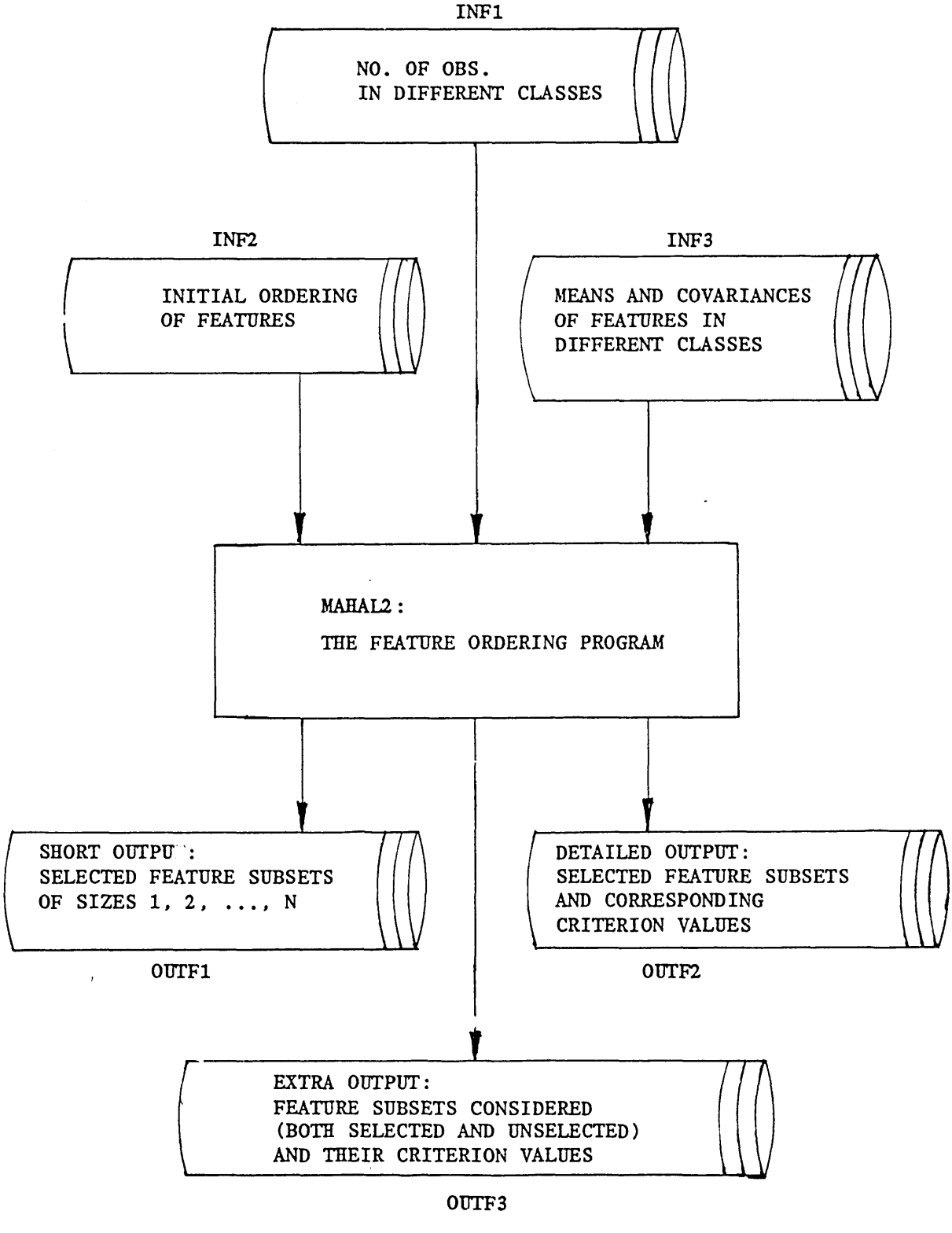


Fig. 5.1 Input and output files of Mahalanobis distance-based step-by-step feature ordering program

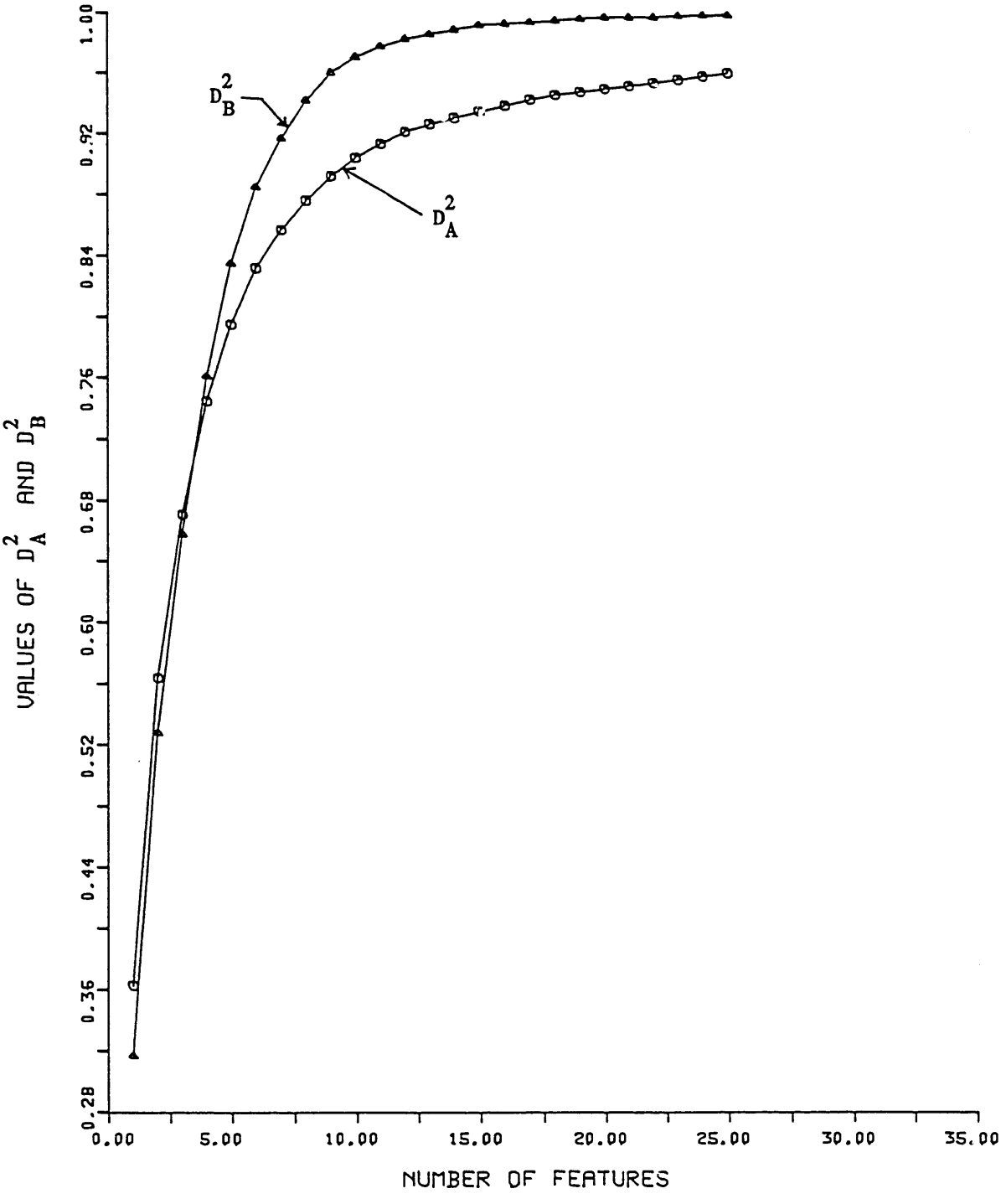


Fig. 5.2 Values of D_A^2 and D_B^2 for different feature subset sizes

Table 5.4a Stepwise feature selection by D_A^2

Step	Feature subset selected	Value of D_A^2
(1)	(2)	(3)
1	13	0.363
2	13, 17	0.564
3	13, 17, 14	0.671
4	13, 17, 14, 43	0.745
5	13, 17, 14, 43, 1	0.795
6	13, 17, 14, 43, 1, 18	0.832
7	13, 17, 14, 43, 1, 18, 8	0.857
8	13, 17, 14, 43, 1, 18, 8, 52	0.876
9	13, 17, 14, 43, 1, 18, 8, 52, 11	0.892
10	13, 17, 14, 43, 1, 18, 8, 52, 11, 44	0.904
11	13, 17, 14, 43, 1, 18, 8, 52, 11, 44, 26	0.913
12	13, 17, 14, ... ,44, 26, 61	0.921
13	13, 17, 14, ... ,26, 61, 12	0.926
14	13, 17, 14, ... ,61, 12, 2	0.930
15	13, 17, 14, ... ,12, 2, 23	0.934
16	13, 17, 14, ... , 2, 23, 9	0.938
17	13, 17, 14, ... ,23, 9, 32	0.942
18	13, 17, 14, ... , 9, 32, 25	0.945
19	13, 17, 14, ... ,32, 25, 31	0.947
20	13, 17, 14, ... ,25, 31, 20	0.949
21	13, 17, 14, ... ,31, 20, 47	0.951
22	13, 17, 14, ... ,20, 47, 16	0.953
23	13, 17, 14, ... ,47, 16, 10	0.955
24	13, 17, 14, ... ,16, 10, 21	0.957
25	13, 17, 14, ... ,10, 21, 22	0.959

Table 5.4b Stepwise feature selection by D_B^2

Step	Feature subset selected	Value of D_B^2
(1)	(2)	(3)
1	13	0.317
2	13, 17	0.528
3	13, 17, 43	0.658
4	13, 17, 43, 14	0.761
5	13, 17, 43, 14, 1	0.835
6	13, 17, 43, 14, 1, 18	0.885
7	13, 17, 43, 14, 1, 18, 23	0.917
8	13, 17, 43, 14, 1, 18, 23, 9	0.942
9	13, 17, 43, 14, 1, 18, 23, 9, 52	0.960
10	13, 17, 43, 14, 1, 18, 23, 9, 52, 10	0.970
11	13, 17, 43, 14, 1, 18, 23, 9, 52, 10, 44	0.977
12	13, 17, 43, ... ,10, 44, 11	0.982
13	13, 17, 43, ... ,44, 11, 32	0.985
14	13, 17, 43, ... ,11, 32, 8	0.988
15	13, 17, 43, ... ,32, 8, 21	0.991
16	13, 17, 43, ... , 8, 21, 22	0.992
17	13, 17, 43, ... ,21, 22, 25	0.993
18	13, 17, 43, ... ,22, 25, 26	0.994
19	13, 17, 43, ... ,25, 26, 12	0.995
20	13, 17, 43, ... ,26, 12, 20	0.996
21	13, 17, 43, ... ,12, 20, 47	0.996
22	13, 17, 43, ... ,20, 47, 16	0.996
23	13, 17, 43, ... ,47, 16, 31	0.997
24	13, 17, 43, ... ,16, 31, 61	0.997
25	13, 17, 43, ... ,31, 61, 2	0.997

In the lower range of values of D_A^2 (and D_B^2) D_A^2 is greater than D_B^2 and as their values increase D_B^2 approaches 1.0 much faster than D_A^2. The reorderings of the above mentioned 25 features obtained by the criteria D_A^2, D_B^2 and D^2 are denoted by $D_A^2(2)$, $D_B^2(2)$ and $D^2(2)$, respectively. They are reported in Table 5.5. '(2)' indicates that the orderings refer to the second stage.

Conformity of the orderings and the recognition results obtained by their use will be the topics of investigation of the next section and the next chapter, respectively.

5.4 Correlation Analysis of Orderings

5.4.1 Correlation Coefficient Used

Kendall's rank correlation coefficient (r_k) and Spearman's rank correlation coefficient (r_s) are widely used for measuring the conformity of two sets of ranks(i.e., orderings). As is the case with the ordinary cross-correlation coefficient, both of these coefficients can have values in the range -1 to +1. A value of 0 indicates that there is no relationship between the two sets of ranks whereas the values of -1 and +1 represent perfect disagreement and perfect agreement, respectively. Several authors [43],[143] have used Spearman's r_s for analyzing the feature orderings. In general r_s is easier to calculate than r_k. But in view of certain distributional as well as computational advantages of Kendall's r_k over Spearman's r_s (pp. 11-12 and p. 46 of [153]) the present analysis will be made on the basis of r_k. Some of the advantages of r_k over r_s are:

Table 5.5
Reordering of 25 features by the Mahalanobis distance-based criteria taking into account the covariances of the features

Rank	$D_A^2(2)$	$D_B^2(2)$	$D^2(2)$
(1)	(2)	(3)	(4)
1	13	13	13
2	17	17	20
3	14	43	17
4	43	14	1
5	1	1	26
6	18	18	18
7	8	23	9
8	52	9	8
9	11	52	21
10	44	10	23
11	26	44	22
12	61	11	12
13	12	32	11
14	2	8	14
15	23	21	25
16	9	22	43
17	32	25	52
18	25	26	10
19	31	12	31
20	20	20	16
21	47	47	44
22	16	16	61
23	10	31	32
24	21	61	47
25	22	2	2

(i) the distribution of r_k is much easier to ascertain,

(ii) the distribution of r_k tends to normality much faster than that of r_s, and

(iii) addition of new members to a ranking does not require a complete recalculation of r_k whereas this will require complete recalculation of r_s.

It is worthwhile mentioning here that when neither coefficient is too close to +1 or −1, r_s is often about 50 percent greater than r_k in absolute value but this will give rise to no difficulty because r_k will be used throughout the present analysis.

Kendall's rank correlation coefficient between two sets of ranks of n values (features) is defined by [153]

$$r_k = \frac{T}{\frac{1}{2} n (n - 1)} \tag{5.18}$$

where

T = number of rank-pairs having same order in both the rankings <u>minus</u> number of rank-pairs having opposite order in the two rankings.

The computer program developed for the computation of r_k is listed in Appendix B 11. Though r_s is not used in the present analysis, its computation is included in the program for completeness.

5.4.2 Significance Tests of Correlation Values

Statistical tests of significance of r_k values are made based on the following properties of r_k:

(i) the standard deviation of r_k is given by

$$\sigma_{r_k} = \sqrt{\frac{2\ (\ 2n + 5\ )}{9n\ (\ n - 1\ )}} \qquad (5.19)$$

and

(ii) the standardized value of r_k, i.e., r_k/σ_{r_k} is distributed, approximately, as a normal deviate (Gaussian variable with mean 0 and standard deviation 1).

At 0.01 level of significance the value of the normal deviate is 2.58. For n = 78 the corresponding value of r_k will be

$$2.58 \times \sqrt{\frac{2\ (\ 2 \times 78 + 5\ )}{9 \times 78 \times (\ 78 - 1\ )}}$$

$$\doteq 0.199$$

Thus, for n = 78, if $|r_k| > 0.199$ then r_k is considered to be significantly different from 0 at 1% level of significance.

Values of r_k between pairs of feature orderings obtained in stage I are shown in Table 5.6. All the tabulated r_k values, except those in the last row showing the correlation coefficients of different orderings with the random ordering R, are larger than 0.199. It can, therefore, be inferred that the feature orderings obtained by

Table 5.6
Values of the Kendall's rank correlation coefficient (r_k) between various pairs of orderings (stage I: n = 78)

	2-class probabilistic criteria				m-class probabilistic criteria				Mahalanobis distance-based criteria		
	$\bar{P}_{e2}$	$\bar{\rho}$	$\bar{\gamma}$	$\bar{J}$	P_e	ρ_m^*	H	B	$D_A^2(1)$	$D_B^2(1)$	$D^2(1)$
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
$\bar{\rho}$	.46										
$\bar{\gamma}$	.56	.41									
$\bar{J}$	.42	.58	.47								
P_e	.42	.53	.42	.51							
ρ_m^*	.51	.56	.42	.46	.52						
H	.39	.50	.39	.48	.58	.54					
B	.35	.31	.35	.39	.41	.37	.41				
$D_A^2(1)$	.34	.44	.33	.29	.45	.38	.41	.23			
$D_B^2(1)$	.31	.34	.42	.36	.35	.31	.31	.32	.53		
$D^2(1)$	.39	.34	.39	.36	.44	.38	.36	.31	.55	.63	
R	-.07	-.15	-.07	-.06	-.09	-.12	-.04	+.13	-.12	-.09	-.17

Theoretical value of r_k at 1% level of significance: 0.199

various probabilistic and Mahalanobis distance-based criteria are in significant agreement with one another. The values of r_k between various feature orderings and the random ordering R are, as expected, small enough to be taken as 0.

In theory the implementation of the m-class Bayesian error probability criterion P_e should lead to optimum feature ordering. Other feature orderings can then be judged by comparing the r_k values between them and the ordering by P_e. From Table 5.6 it can be seen that with P_e, H has the highest r_k value of 0.58, $\bar{\rho}$ having the next highest r_k value of 0.53. One would, therefore, expect the features selected by H and $\bar{\rho}$ to be more effective than those selected by the other criteria. However, in view of the fact that the interrelationships between the features have not been taken into account in stage I, it is difficult to make such a strong recommendation on the basis of so little differences between various r_k values.

Values of r_k between pairs of feature orderings obtained by the implementation of D^2-based criteria in stage II are shown in Table 5.7. As mentioned earlier, in stage II the top 25 features selected by D_A^2 criterion in stage I were the initial set. Column 1 of the table (with the heading $D_A^2(1)$) refers to the correlation values with this initial ordering. For n = 25 the theoretical value of r_k at 1% level of significance is 0.368. Four out of the six tabulated r_k values are greater than 0.368. The corresponding feature orderings are therefore in significant agreement with each other at 1% level. The largest r_k value of 0.513 is observed between $D_A^2(2)$ and $D_B^2(2)$, an indication of more agreement between them compared to any other pair

Table 5.7
Values of the Kendall's rank correlation coefficient (r_k) between pairs of feature orderings obtained by Mahalanobis distance-based criteria (stage II: n = 25)

	$D_A^2(1)$	$D_A^2(2)$	$D_B^2(2)$
(1)	(2)	(3)	(4)
$D_A^2(2)$	.373		
$D_B^2(2)$	.247	.513	
$D^2(2)$	.433	.260	.387

Theoretical value of r_k at 1% level of significance: 0.368

of orderings. This is expected because both of them are normalized versions of the Mahalanobis distance. Though some of the other r_k values are significant at 1% level, a general observation is that the correlation values are not large enough to ascertain close agreement between the feature orderings.

5.5 Some Remarks on Feature Ordering Experiments

The feature ordering experiments described in sections 5.2 and 5.3 suffer from certain design limitations. For the implementation of a probabilistic criterion (see section 5.2) the features were treated individually. For the selection of the best individual feature this procedure may be applied. But, since the interactions between the features are ignored, the above procedure is likely to fail in a situation where one is interested in selecting a subset of features containing more than one feature. For the implementation of the Mahalanobis distance-based criteria (see section 5.3) a step-by-step feature selection procedure was applied. This procedure does not lead to optimum selection of features because, for feature subsets of size greater than 1, all the possible feature subsets are not considered for evaluation. Therefore, in either of the above two approaches, it is difficult to draw conclusions regarding the comparative effectiveness of different feature evaluation criteria based on the comparison of the feature orderings obtained by the use of these criteria. It is therefore suggested that for the selection of features in a practical pattern recognition problem care should be taken to reduce the effects of the above limitations.

CHAPTER 6

RECOGNITION EXPERIMENTS

6.1 Introduction

In this chapter recognition experiments are conducted using features selected in the previous chapter. Analysis of recognition accuracies obtained by the use of features selected by different feature evaluation criteria enables a comparative assessment to be made of these criteria.

In section 6.2 the classification criterion used in the recognition experiments is stated. Section 6.3 deals with the experimental procedures and the recognition results. Two sets of experiments are conducted, one with the same training and test data and the other following the leave-one-out principle of Lachenbruch and Mickey [154]. These two sets of experiments are the subjects of study of subsections 6.3.1 and 6.3.2, respectively. In section 6.4 some two-class recognition experiments are conducted taking the numerals '3' and '5' as the two classes. In section 6.5 the experimental results are summarized.

6.2 Classification Criterion Used

According to the Bayesian decision procedure stated in chapter 2 (section 2.1), a pattern with the value of its feature vector X = x is assigned to class C_i, i=1,2,...,m if

$$\pi_i p(x|C_i) > \pi_j p(x|C_j), \qquad j=1,2,\ldots,m; \qquad j \neq i \tag{6.1}$$

In other words, the Bayesian procedure assigns x to C_i if $R_i > R_j$ for all $j \neq i$ where

$$R_i = \pi_i p(x|C_i), \qquad i=1,2,\ldots,m \tag{6.2}$$

If the features $X_1, X_2, \ldots, X_n$ are independent then (6.2) reduces to

$$R_i = \pi_i \prod_{k=1}^{n} p(x_k|C_i), \quad i=1,2,\ldots,m \tag{6.3a}$$

Because of the monotonicity property of the logarithmic function, maximization of R_i given by (6.3a) is equivalent to maximization of R_i' given by

$$R_i' = \ln \pi_i + \sum_{k=1}^{n} \ln [\, p(x_k|C_i) \,], \quad i=1,2,\ldots,m \tag{6.3b}$$

In the above expressions π_i and $p(x_k|C_i)$ denote the a'priori probability of the class C_i and the ith class-conditional probability density function of the feature X_k at $X_k = x_k$, respectively.

In real-life applications the criterion R_i suffers from the drawback that its value becomes 0 if (the estimate of) one of the n $p(x_k|C_i)$'s, k=1,2,...,n is equal to 0 for all i=1,2,...,m. This means that inclusion of only one useless feature would be enough to spoil the applicability of the criterion R_i. Similar computational problem arises if R_i' is used because, for $p(x_k|C_i) = 0$, the value of $\ln[p(x_k|C_i)]$ becomes undefined. The problem, however, can be tackled by assigning a reasonably small value to $\ln[p(x_k|C_i)]$.

In the present numeral recognition experiments using the whole data set for both training and testing the criterion R_i was applied and in the case of leave-one-out principle R_i' was used. The assumption of independence of features, implicit in the use of R_i or R_i' as the classification criterion, resulted in the simplification of implementation of the criterion and reduction in storage and computational requirements.

In the two sets of classification experiments to be dealt with in the next section each of the 1000 numeral samples was classified in one of 13 classes. For further analysis of the classification results the classes representing the same numeral were combined to get back to the original 10 class situation. A confusion matrix of order 10 x 10 was then obtained whose (i,j)th element, f_{ij}, was the number of observations actually belonging to the numeral category i but recognized as belonging to the category j (i=0,1,2,...,9; j=0,1,2,...,9). Recognition accuracy was then computed using the

following equation:

$$\text{Correct classification rate} = \frac{\sum_{i=0}^{9} f_{ii}}{\sum_{i=0}^{9}\sum_{j=0}^{9} f_{ij}} \times 100 \text{ (per cent)} \qquad (6.4)$$

Since all the 1000 numerals underwent the classificatory analysis, the denominator of (6.4) is equal to 1000.

6.3 Experiments and Results

6.3.1 Recognition with the Same Training and Test Data

Each of the 1000 numeral samples was recognized using the R_i criterion (Equation 6.3a). Implementation of the recognition procedure was quite straightforward. The estimates of the a'priori probabilities of the 13 classes under consideration were 0.100, 0.080, 0.020, 0.100, 0.100, 0.072, 0.028, 0.100, 0.100, 0.060, 0.040, 0.100 and 0.100 (based on Table 5.1). The estimates of the class-conditional probability densities, in fact discrete approximations to them, were those obtained in chapter 5 (section 5.2.1). Given a set of features on the basis of which to recognize the numerals, for each numeral the intervals in which the values of its above features were lying were determined. The estimates of class-conditional probabilities corresponding to these intervals

and the estimates of a'priori probabilities were then put in the equation (6.3a) to compute the R_i values for i=1,2,...,13. The class associated with the maximum R_i value was then considered as the recognized class. After completion of the recognition procedure for all the 1000 numeral samples the confusion matrix was generated which was then followed by the computation of the correct classification rate using the equation (6.4).

A program developed for the implementation of the above procedure is given in Appendix B 12. The program works for a number of feature subsets. From a list of features, supplied as an input file, the program obtains information on which features to use and then recognizes each numeral sample on the basis of the values of these features.

The recognition results obtained by using various feature orderings reported in the previous chapter will now be analyzed. Percentages of correct classification achieved by using the features selected by various probabilistic criteria are shown in Table 6.1. Feature subset sizes considered were 1,2,...,15,20,...,75 and 78. For features used in arriving at the results of Table 6.1 refer to Table 5.2 (chapter 5). Out of the eight probabilistic criteria experimented five chose the feature number 13 to be the best feature. Use of this feature led to a recognition accuracy of 29.5%. P_e, ρ_m^* and B selected the features 18, 47 and 43, respectively. The corresponding recognition accuracies were 27.6%, 24.5% and 28%, respectively. Recognition results obtained by the use of features arranged by D^2-based criteria (for feature orderings see Table 5.3) are shown in Table 6.2. All the D^2-based criteria, namely, D_A^2, D_B^2 and

Table 6.1
Percentages of correct recognition with 'same training and test data' and using features selected by probabilistic criteria

Serial number	No. of features	Feature ordering used							
		$\bar{P}_{e2}$	$\bar{\rho}$	$\bar{\gamma}$	$\bar{J}$	P_e	ρ_m^*	H	B
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	1	29.5	29.5	29.5	29.5	27.6	24.5	29.5	28.0
2	2	45.4	45.4	45.4	45.4	38.6	39.2	44.7	39.2
3	3	57.9	54.7	57.9	54.7	56.0	45.9	56.0	50.9
4	4	67.2	58.8	63.7	63.2	67.3	56.6	62.8	63.2
5	5	73.2	66.0	67.5	66.0	73.8	62.7	69.5	72.9
6	6	79.6	71.2	77.4	75.7	80.7	74.7	76.9	74.3
7	7	82.2	76.0	82.2	81.2	82.3	78.2	81.2	76.3
8	8	84.7	85.0	85.0	85.0	83.8	80.4	84.5	80.9
9	9	88.4	87.5	86.1	86.1	85.0	83.8	87.5	84.4
10	10	88.4	88.3	87.5	88.3	88.6	85.2	88.4	87.6
11	11	89.0	89.3	88.9	90.7	88.9	88.1	88.3	87.9
12	12	90.8	90.2	90.4	91.2	90.0	88.3	88.8	89.9
13	13	91.4	91.9	91.5	91.7	91.8	91.0	89.0	90.8
14	14	91.7	91.7	92.6	91.7	92.9	92.5	90.5	91.4
15	15	92.6	92.2	93.3	92.2	93.0	93.4	91.8	91.5
16	20	95.3	94.9	95.8	94.4	94.6	95.5	95.5	95.2
17	25	96.4	96.3	96.3	96.3	96.2	95.5	96.1	96.1
18	30	97.4	97.0	97.4	97.0	96.9	97.0	96.9	96.8
19	35	98.0	98.2	98.0	98.2	98.3	98.3	98.2	98.5
20	40	98.3	98.3	98.4	98.3	98.4	98.4	98.3	98.3
21	45	98.7	98.7	98.7	98.7	98.8	98.8	98.8	98.6
22	50	98.7	98.7	98.8	98.7	98.7	98.8	98.7	98.7
23	55	98.9	98.9	98.8	98.9	98.8	98.8	98.9	98.8
24	60	99.0	99.0	99.0	99.0	99.0	99.0	99.0	98.9
25	65	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
26	70	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
27	75	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
28	78	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0

Table 6.2
Percentages of correct recognition with 'same training and test data' and using features selected by D^2-based criteria in stage I (also included the recognition scores for a random ordering, R, of features)

Serial number	No. of features	Feature ordering used			
		$D_A^2(1)$	$D_B^2(1)$	$D^2(1)$	R
(1)	(2)	(3)	(4)	(5)	(6)
1	1	29.5	29.5	29.5	13.2
2	2	39.7	39.7	42.6	19.4
3	3	51.5	48.5	53.8	21.3
4	4	63.2	58.1	65.4	31.5
5	5	66.0	66.0	71.3	41.3
6	6	71.1	71.1	77.8	50.1
7	7	76.7	78.9	80.8	50.9
8	8	84.0	81.8	83.3	58.7
9	9	86.7	86.8	84.3	59.0
10	10	89.1	89.1	85.4	59.8
11	11	90.7	90.4	86.7	64.0
12	12	90.7	91.4	89.2	64.1
13	13	92.3	91.6	91.1	64.0
14	14	93.3	92.7	91.5	71.1
15	15	94.1	93.1	93.1	70.8
16	20	95.3	95.5	94.8	85.5
17	25	96.3	96.3	96.3	89.4
18	30	97.0	97.1	97.4	90.8
19	35	98.2	98.4	98.4	94.5
20	40	98.1	98.1	98.1	96.4
21	45	98.8	98.8	98.8	96.6
22	50	98.7	98.7	98.7	98.0
23	55	98.8	98.8	98.8	98.4
24	60	98.9	98.9	98.9	98.2
25	65	98.9	98.9	98.9	98.3
26	70	99.0	99.0	99.0	98.7
27	75	99.0	99.0	99.0	98.8
28	78	99.0	99.0	99.0	99.0

D^2, selected the feature 13, the feature leading to the recognition accuracy of 29.5% which was maximum of all the entries shown in the first row of Table 6.1 and Table 6.2. In other words, all the D^2-based criteria selected the single best feature. As expected, with increase in the number of features the recognition accuracy increased, the rate of increase going down for a larger number of features. This was true irrespective of whatever feature ordering was used. By using only 14 features out of 78 features more than 90% correct recognition could be obtained. With most of the orderings, use of 20 features led to a recognition accuracy of more than 95%. To achieve a recognition accuracy of 99% at least 60 features were needed. Even by using all the 78 features more than 99% accuracy could not be obtained. Compared to R, a random ordering, all other orderings led to a much better recognition performance. This is, indeed, an indication of the usefulness of the feature ordering criteria in genaral. From Tables 6.1 and 6.2 it may be noted that the recognition accuracies do not vary much from ordering to ordering. This is a very strong point in favour of using the D^2-based criteria because, for the evaluation of individual features by them, one needs to estimate only the means and the standard deviations of the features. Amongst the D^2-based criteria, D_A^2 performed better than D_B^2. For smaller number of features (upto 7 features) D^2 performed better than both D_A^2 and D_B^2 but for more than 7 features, in most of the cases, performance of D_A^2 was better than that of D^2.

The recognition results obtained by using various subsets of 25 features reordered in stage II, and taking into account the covariances between the features, will now be analyzed. For the reorderings obtained by D_A^2, D_B^2 and D^2 criteria refer to Table 5.5.

Corresponding recognition results, for feature subsets of sizes 1,2,...,20 and 25, are presented in Table 6.3. Both D_A^2 and D_B^2 performed better than D^2 for all the feature subset sizes. This conformed well with the expectation that D_A^2 and D_B^2, being normalized criteria, they should perform better than D^2 in a multiclass situation. Fig. 6.1 illustrates the improvement gained in stage II, over stage I, with D_A^2 as the evaluation criterion. For a combination of two features the recognition accuracy increased from 39.7% to 45.4%. For three features the corresponding values were 51.5% and 58.6%, respectively. Thus, a set of 3 features selected taking into account the covariances between features gave rise to 7.1% more recognition accuracy compared to the set of 3 individually best features. This is, indeed, a remarkable improvement. Improvements were observed for higher number of features also. From Tables 6.1 and 6.3 it can be seen that $D_A^2(2)$, the ordering by D_A^2 criterion in stage II, gave rise to much better recognition scores than those by the probabilistic criteria. For most of the feature subset sizes $D_B^2(2)$ also performed better than the probabilistic criteria. One may argue that the probabilistic criteria were applied without any consideration to the interactions between the features. But it sh uld be noted that the storage requirement for their implementation was quite high because one had to consider a number of intervals, for each feature, for the estimation of class-conditional probabilities.

On the basis of the experimental results with the same training and test data one can therefore consider D_A^2 and D_B^2 to be two useful feature evaluation criteria.

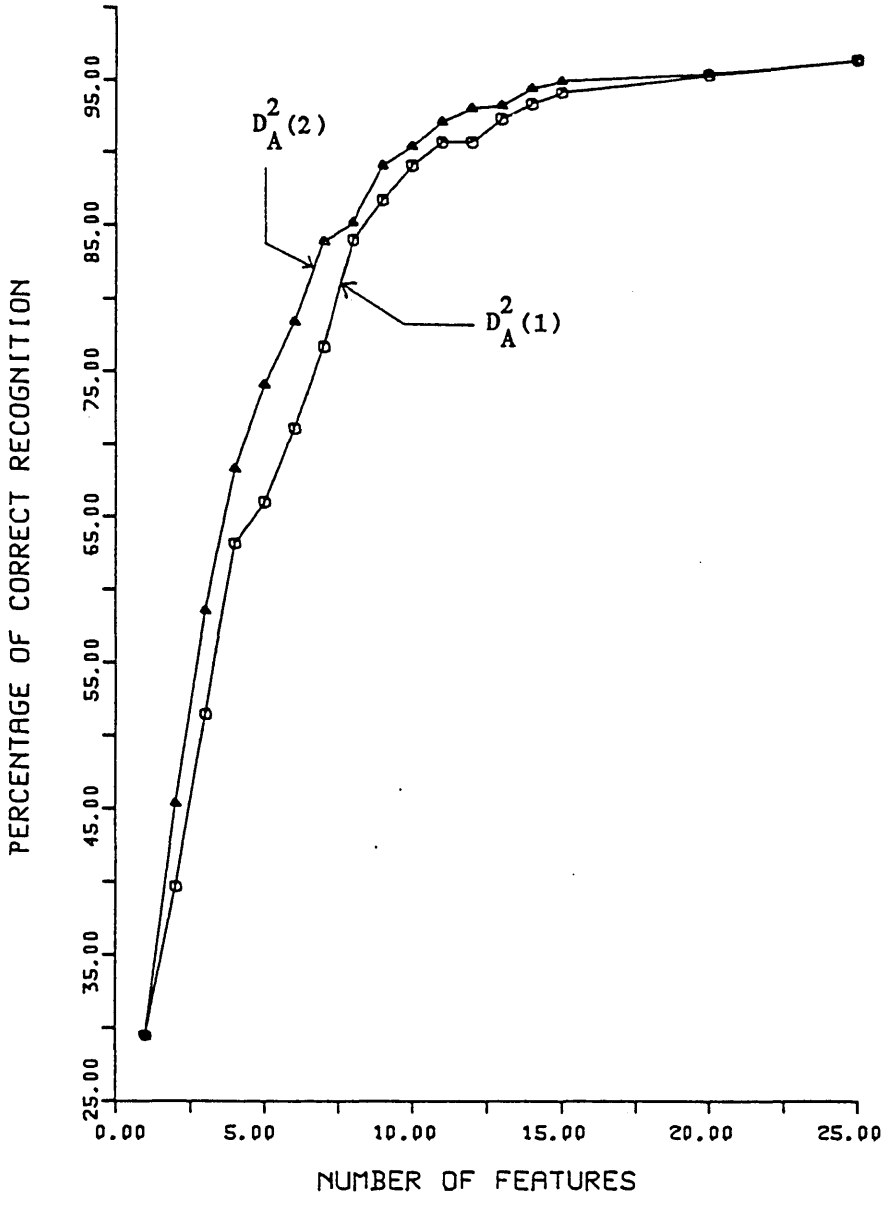


Fig. 6.1 Recognition accuracies in stages I and II, with the same training and test data, for features selected by D_A^2 criterion

Table 6.3
Percentages of correct recognition with 'same training and test data' and using features reordered by D^2-based criteria in stage II

Serial number	No. of features	Feature ordering used		
		$D_A^2(2)$	$D_B^2(2)$	$D^2(2)$
(1)	(2)	(3)	(4)	(5)
1	1	29.5	29.5	29.5
2	2	45.4	45.4	42.6
3	3	58.6	56.0	55.6
4	4	68.3	68.3	66.4
5	5	74.1	74.1	69.5
6	6	78.4	78.4	75.1
7	7	83.9	81.4	78.5
8	8	85.2	86.5	84.0
9	9	89.1	87.5	87.2
10	10	90.4	90.3	88.2
11	11	92.1	91.6	90.0
12	12	93.0	93.0	90.8
13	13	93.2	93.4	91.3
14	14	94.4	93.2	92.5
15	15	94.9	94.4	92.5
16	20	95.4	95.9	95.2
17	25	96.3	96.3	96.3

6.3.2 Recognition by Leave-One-Out Principle

The wellknown drawback in using the same data set for both training and testing is that it has a tendency towards overestimating the recognition accuracy [91],[154]. If the available data set is large then this drawback can be avoided by dividing the data into separate training and test sets. Since there were only 1000 observations for the experimentation of a 13-class problem with 78 features, the leave-one-out principle was adopted. Implementation of this approach helps in two ways: (i) it gives a more realistic estimate of the recognition accuracy and (ii) it enables one to study the severity of the problem of using the same training and test data.

A program developed for the implementation of the leave-one-out principle is listed in Appendix B 13. The major functional steps involved in the program will now be described. The estimates of the a'priori probabilities are the same as those used previously in the case of same training and test data. To start with, consider the class-conditional probabilities of different features in different classes estimated on the basis of the whole data set. These probability estimates are modified before using them for the recognition of each numeral. The modification procedure used will now be described using the same notations as those in the program.

Let M be the number of classes and N be the number of features. Let SIZE(I),I=1,2,...,M be the number of observations from the I-th class in the whole data set. Suppose P(K,I,J) is the proportion of values in the whole data set, taken by the K-th feature in the I-th class, which lie in the J-th interval. Then P(K,I,J)'s are the starting estimates of the class-conditional probabilities.

Suppose, in a cycle of recognition process, it is required to recognize an observation which comes from the class INEW and whose feature values are XNEW(1), XNEW(2), ..., XNEW(N). Suppose the observation recognized at the previous cycle of recognition came from the class IOLD and its feature values were XOLD(1), XOLD(2), ..., XOLD(N). For each of the N features determine the interval containing XNEW(K), K=1,2,...,N. Suppose it is JNEW(K). Suppose the interval containing XOLD(K) is JOLD(K). Then the previous P(K,I,J)'s are modified to new P(K,I,J)'s by making the following substitutions:

$$P(K,IOLD,J) = \begin{cases} \dfrac{P(K,IOLD,J)\ .\ [SIZE(IOLD) - 1]}{SIZE(IOLD)} & \text{for } J \neq JOLD(K) \\ \dfrac{P(K,IOLD,J)\ .[\ SIZE(IOLD) - 1] + 1}{SIZE(IOLD)} & \text{for } J = JOLD(K) \end{cases} \tag{6.5a}$$

and

$$P(K,INEW,J) = \begin{cases} \dfrac{P(K,INEW,J)\ .\ SIZE(INEW)}{SIZE(INEW) - 1} & \text{for } J \neq JNEW(K) \\ \dfrac{P(K,INEW,J)\ .\ SIZE(INEW) - 1}{SIZE(INEW) - 1} & \text{for } J = JNEW(K) \end{cases} \tag{6.5b}$$

It is worth mentioning here that while recognizing the first observation, only (6.5b) is implemented and for all other observations both (6.5a) and (6.5b) are implemented. Once the P(K,I,J)'s are modified as above, the observation under consideration is recognized using the criterion R' of equation (6.3b). Depending on the number of features to be used decision is made about which features to include in the computation of R'. As before, the class corresponding to the maximum R'-value is then decided to be the recognized class.

Percentages of correct classification achieved by using the features selected by the probabilistic criteria and following the leave-one-out principle are shown in Table 6.4. All the entries in Table 6.4 are considerably lower than the corresponding entries in Table 6.1. This indicates that the use of the same data for both training and testing introduces a considerable positive bias in the recognition accuracy. This point is illustrated in Fig. 6.2 wherein the recognition accuracies achieved by the two methods using the features selected by the m-class Bayesian error criterion P_e are plotted. For the other probabilistic criteria diagrammatic representations of the results achieved by the two methods are not given in the thesis, but from a comparison of Tables 6.1 and 6.4 it can be seen that the use of leave-one-out principle resulted in similar reduction in recognition accuracy for all of them.

The results of Table 6.4 will now be analyzed to get a comparative picture of different probabilistic criteria. It is interesting to note that the recognition scores achieved by the use of features selected by different criteria compared with one another, more or less in the same way, as they did in the case of same training

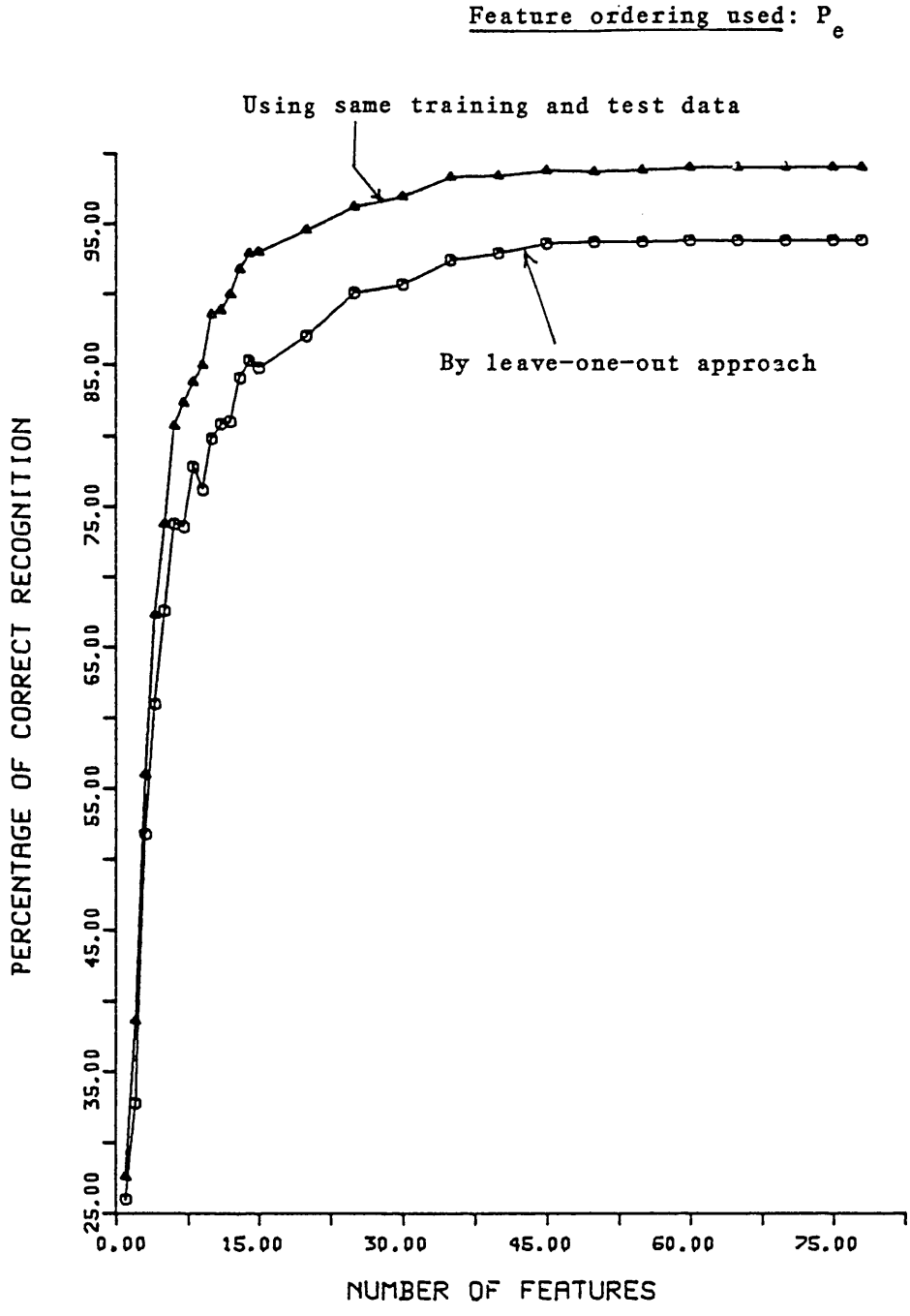


Fig. 6.2 Recognition accuracies by 'same training and test data approach' and 'leave-one-out approach' for features selected by P_e criterion

Table 6.4
Percentages of correct recognition by leave-one-out method and using features selected by probabilistic criteria

Serial number	No. of features	Feature ordering used							
		$\bar{P}_{e2}$	$\bar{\rho}$	$\bar{\gamma}$	$\bar{J}$	P_e	ρ_m^*	$\bar{H}$	B
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	1	28.3	28.3	28.3	28.3	26.0	23.3	28.3	25.6
2	2	41.0	41.0	41.0	41.0	32.8	37.2	42.4	37.2
3	3	53.7	49.5	53.7	49.5	51.8	42.3	50.9	46.7
4	4	63.0	54.1	57.8	58.2	61.0	52.2	57.0	60.3
5	5	67.0	60.4	61.3	60.4	67.6	58.3	63.4	68.5
6	6	72.5	64.2	71.0	69.6	73.8	70.0	69.8	68.9
7	7	75.7	67.7	75.7	73.1	73.6	71.1	73.1	70.6
8	8	77.9	77.1	77.1	77.1	77.8	73.3	77.3	74.5
9	9	82.1	80.4	78.2	78.2	76.2	77.2	80.4	77.4
10	10	82.0	80.3	79.2	80.3	79.8	77.6	80.6	80.4
11	11	81.3	81.5	81.9	83.0	80.8	80.9	80.7	80.0
12	12	83.4	83.2	83.7	83.7	81.0	82.4	80.1	81.6
13	13	85.7	85.0	85.4	85.4	84.1	84.3	81.8	82.4
14	14	85.4	84.6	85.8	85.3	85.3	85.4	83.1	82.5
15	15	85.7	84.8	86.3	84.8	84.8	86.7	84.9	84.2
16	20	89.2	89.4	90.5	88.8	87.1	89.8	89.6	87.5
17	25	89.9	90.3	90.2	90.3	90.1	89.4	90.0	90.9
18	30	91.3	91.5	92.0	91.5	90.7	90.8	91.1	91.0
19	35	92.6	92.1	92.6	92.1	92.4	92.2	92.1	92.6
20	40	93.0	93.0	92.9	93.0	92.9	92.6	93.0	93.0
21	45	93.5	93.5	93.5	93.5	93.6	93.3	93.6	93.5
22	50	93.5	93.7	93.6	93.7	93.7	93.6	93.8	93.9
23	55	93.7	93.7	93.8	93.7	93.7	93.8	93.6	93.7
24	60	93.8	93.8	93.8	93.8	93.8	93.8	93.8	93.6
25	65	93.8	93.7	93.7	93.8	93.8	93.8	93.8	93.8
26	70	93.8	93.8	93.8	93.8	93.8	93.8	93.8	93.8
27	75	93.8	93.8	93.8	93.8	93.8	93.8	93.8	93.8
28	78	93.8	93.8	93.8	93.8	93.8	93.8	93.8	93.8

and test data. For most of the feature subset sizes $\bar{P}_{e2}$ performed better than P_e. Performances of the two criteria were, in general, better than those of the other indirect criteria. Amongst others, $\bar{\gamma}$ performed better than the rest. For smaller number of features (upto 5) ρ_m^* showed the worst performance but for higher number of features its performance was very similar to those of others.

Recognition results, by leave-one-out method, using features selected by D^2-based criteria in stage I are shown in Table 6.5. Comparison of Tables 6.2 and 6.5 shows that the leave-one-out principle resulted in remarkable decrease in recognition accuracy. As in the case of probabilistic criteria, the extent of the decrease was greater for higher numbers of features. The nature of reduction for various feature set sizes is illustrated in Fig. 6.3 wherein the plotted recognition rates were obtained by using the features selected by the criterion D_A^2. Recognition results obtained by using the D^2-based criteria compared well with the results of most of the probabilistic criteria. It was observed that for smaller number of features their performances were not as good as those of $\bar{P}_{e2}$ and P_e but the gap was negligible for 10 or more features.

On the basis of the recognition results shown in Table 6.5 it is difficult to compare the performances of the three D^2-based criteria. For a number of features upto 8, D^2 performed slightly better than both D_A^2 and D_B^2 but the situation was reversed when the number of features was between 9 and 15. For more features the differences between them were small. This was in fact the case with the probabilistic criteria also.

Feature ordering used: $D_A^2(1)$

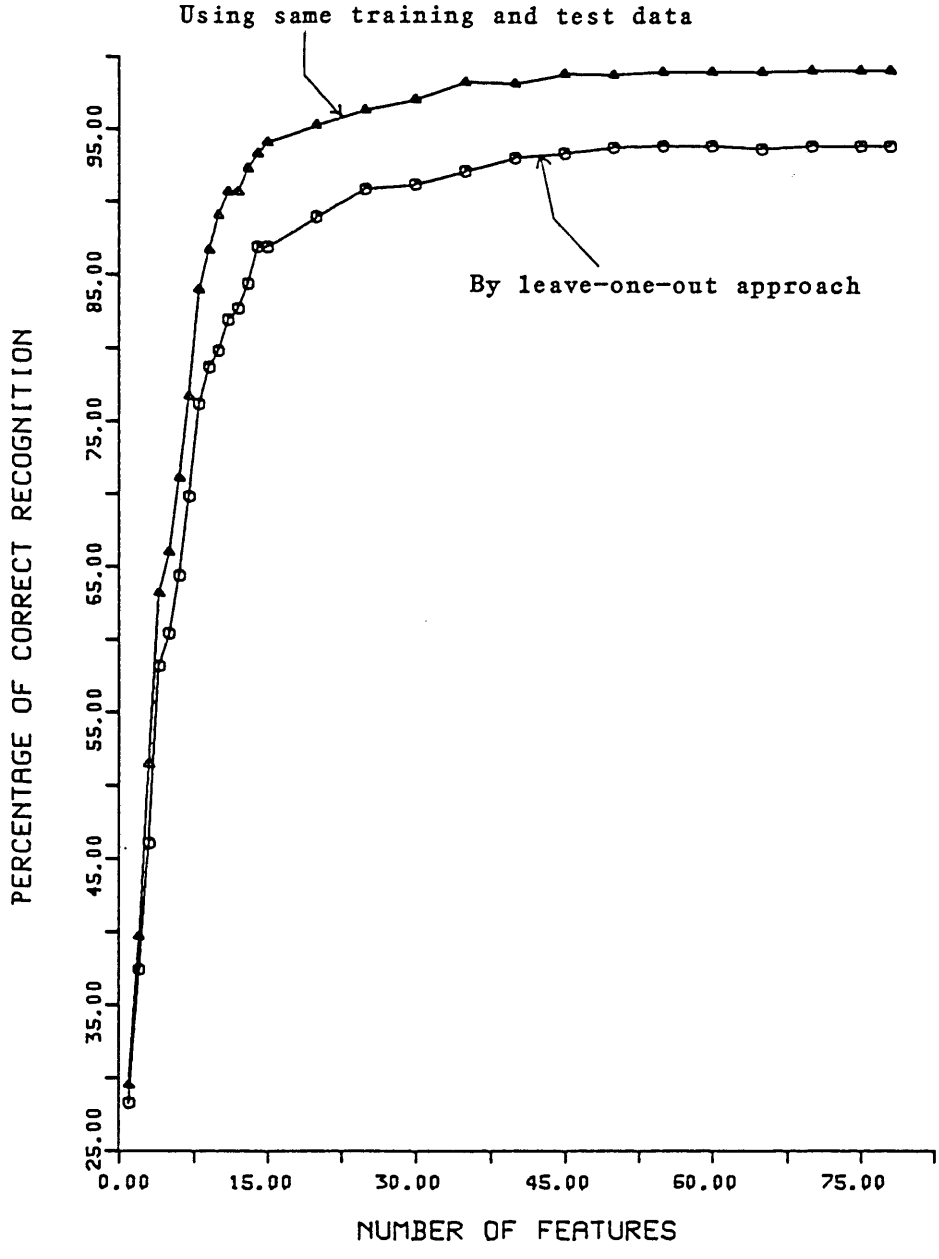


Fig. 6.3 Recognition accuracies by 'same training and test data approach' and 'leave-one-out approach' for features selected by D_A^2 criterion in stage I

Table 6.5
Percentages of correct recognition by leave-one-out method and using features selected by D^2-based criteria in stage I (also included the recognition scores for a random ordering, R, of features)

Serial number	No. of features	Feature ordering used			
		$D_A^2(1)$	$D_B^2(1)$	$D^2(1)$	R
(1)	(2)	(3)	(4)	(5)	(6)
1	1	28.3	28.3	28.3	12.8
2	2	37.4	37.4	38.8	18.8
3	3	46.1	46.4	48.6	20.1
4	4	58.2	51.8	61.0	29.6
5	5	60.4	60.4	65.8	38.0
6	6	64.4	64.4	71.1	43.7
7	7	69.8	70.5	75.0	43.5
8	8	76.2	75.0	77.4	51.7
9	9	78.7	79.6	78.1	51.3
10	10	79.8	79.8	78.3	51.6
11	11	81.9	82.3	78.8	56.4
12	12	82.7	84.0	81.8	56.9
13	13	84.4	84.3	83.6	56.9
14	14	86.9	84.1	83.9	62.5
15	15	86.9	85.0	85.0	62.4
16	20	89.0	89.7	89.9	77.4
17	25	90.9	91.3	91.3	81.0
18	30	91.2	91.5	92.0	84.2
19	35	92.1	92.3	92.3	88.7
20	40	93.0	93.0	93.0	91.4
21	45	93.3	93.3	93.3	90.7
22	50	93.7	93.7	93.7	92.0
23	55	93.8	93.8	93.8	92.3
24	60	93.8	93.8	93.8	92.3
25	65	93.6	93.6	93.6	92.4
26	70	93.8	93.8	93.8	93.1
27	75	93.8	93.8	93.8	93.5
28	78	93.8	93.8	93.8	93.8

The recognition results obtained by using the features selected in stage II will now be analyzed. The results are shown in Table 6.6. The superiority of D_A^2 and D_B^2 over D^2 is evident from the recognition scores. For pictorial representation of the performances of these three criteria refer to Fig. 6.4. D_A^2 and D_B^2 produced better results than D^2 for all sizes of feature subsets. Results by D_A^2 and D_B^2 did not differ much. In general, D_A^2 performed better than D_B^2.

The improvement gained by the implementation of stage II is shown in Fig. 6.5 wherein the recognition scores obtained by D_A^2 in stages I and II are plotted for various feature subset sizes upto 25. $D_A^2(2)$ resulted in much better performance than $D_A^2(1)$. For 5 features the recognition accuracy increased from 60.4% to 69.6%, a remarkable increase of 9.2%. For higher number of features the increase was less. This is expected because the selection was made from a set of 25 features only.

In Fig. 6.6 the recognition scores for features selected by P_e and for features selected by D_A^2 in stage II are plotted. It is very encouraging to note that the recognition scores of $D^2{}_A(2)$ were better than those of P_e. As mentioned earlier, P_e performed slightly better than $D^2{}_A(1)$. But the implementation of stage II made such remarkable improvements that $D_A^2(2)$ gave better results than P_e.

6.4 Results of a Set of 2-Class Experiments

Some 2-class recognition experiments were conducted with the numerals '3' and '5' as the two classes. Selection of the numerals was based on two reasons. Firstly, from an analysis of the confusion

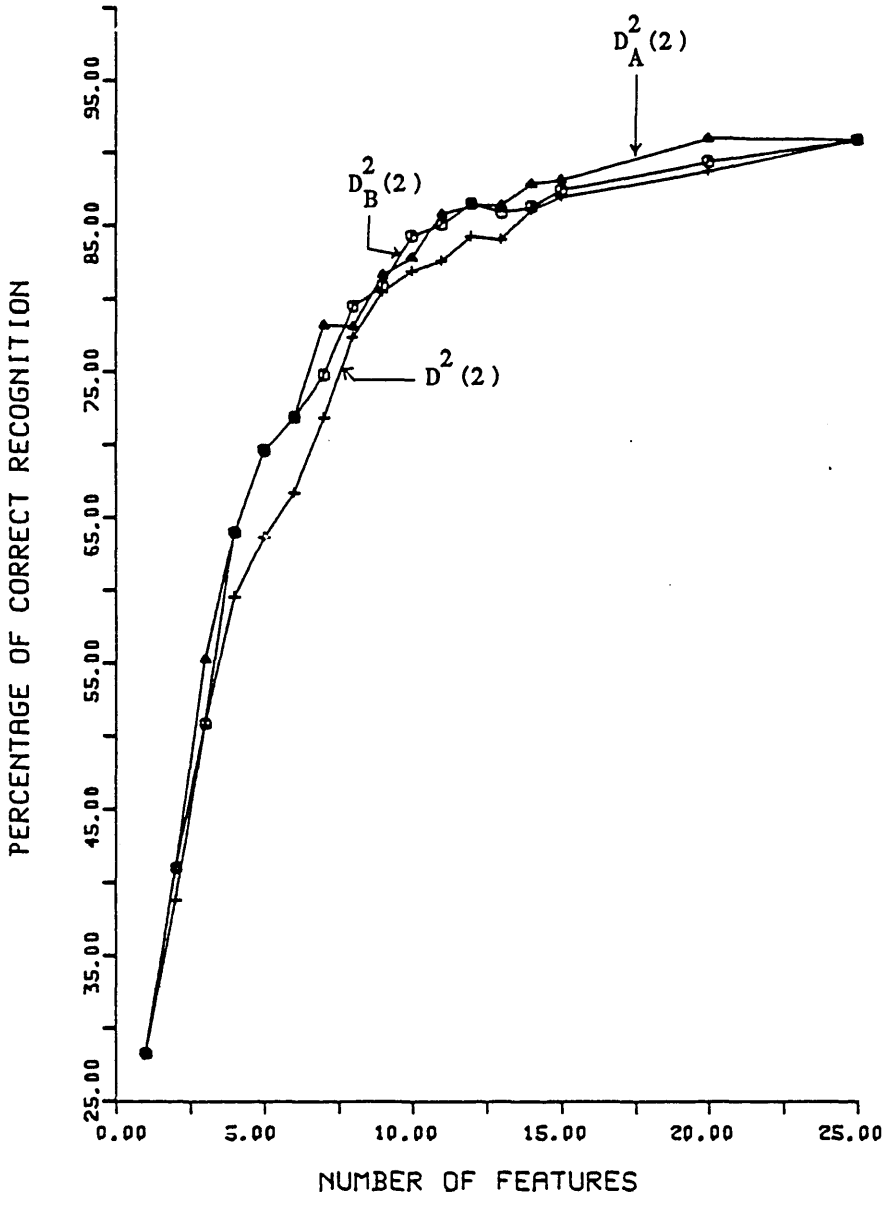


Fig.6.4 Recognition accuracies for features selected by D^2-based criteria in stage II following leave-one-out approach

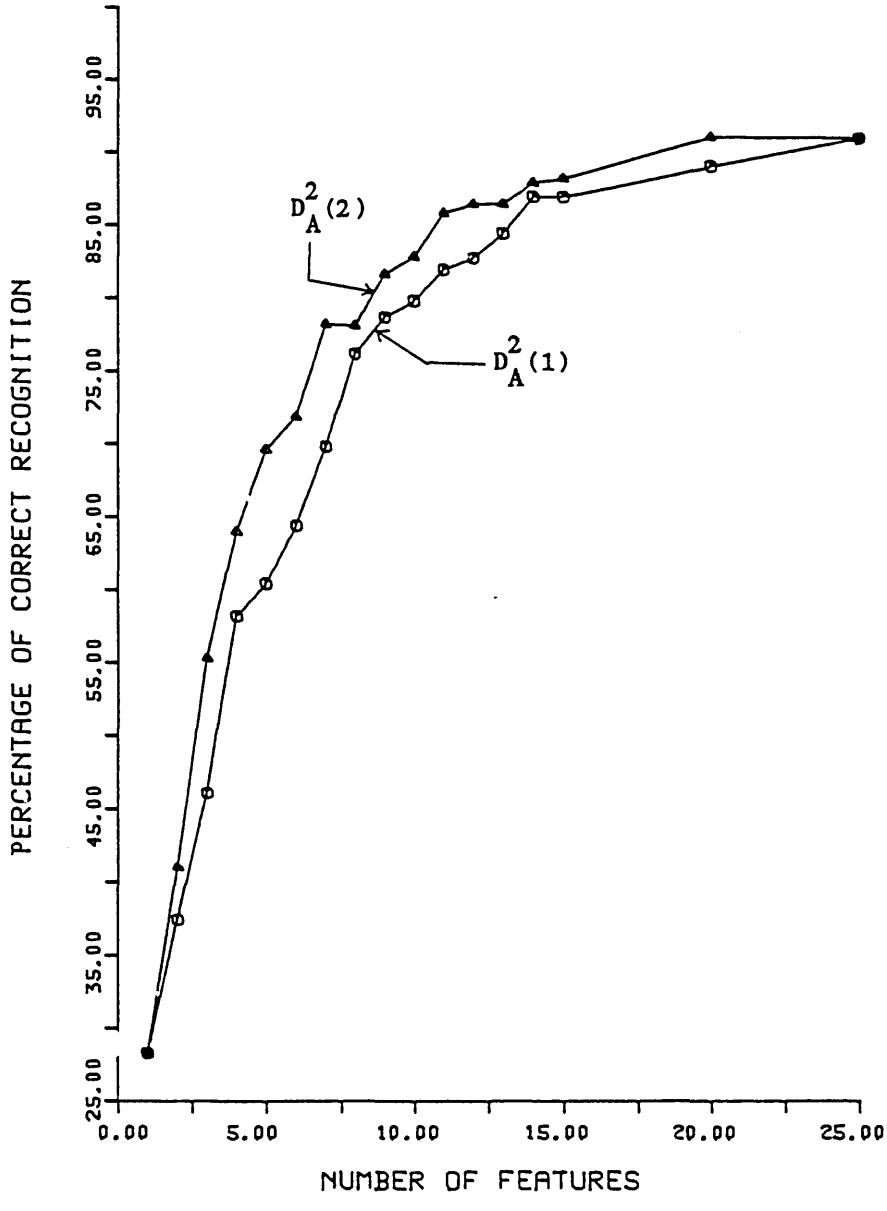


Fig. 6.5 Recognition accuracies in stages I and II, by leave-one-out method, for features selected by D_A^2 criterion

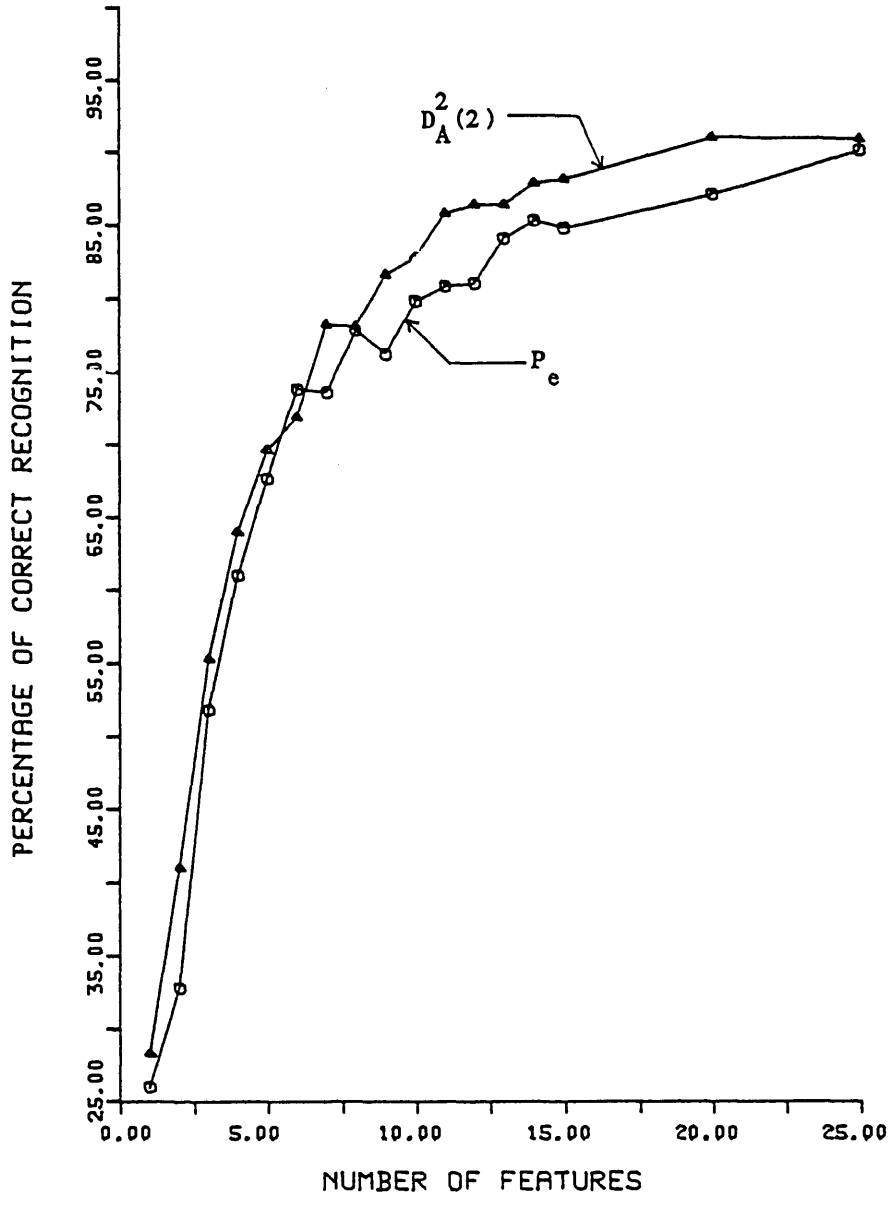


Fig.6.6 Recognition accuracies, by leave-one-out method, for features selected by P_e and D_A^2 (in stage II)

Table 6.6
Percentages of correct recognition by leave-one-out method and using features reordered by D^2-based criteria in stage II

Serial number	No. of features	Feature ordering used		
		$D_A^2(2)$	$D_B^2(2)$	$D^2(2)$
(1)	(2)	(3)	(4)	(5)
1	1	28.3	28.3	28.3
2	2	41.0	41.0	38.8
3	3	55.3	50.9	50.8
4	4	64.0	64.0	59.6
5	5	69.6	69.6	63.7
6	6	71.9	71.9	66.7
7	7	78.2	74.8	71.9
8	8	78.1	79.5	77.4
9	9	81.6	80.9	80.5
10	10	82.8	84.3	81.9
11	11	85.8	85.1	82.6
12	12	86.4	86.5	84.3
13	13	86.4	85.9	84.1
14	14	87.9	86.3	86.1
15	15	88.2	87.5	87.0
16	20	91.0	89.4	88.8
17	25	90.9	90.9	90.9

matrices obtained in the experiments conducted in the previous section it was found that the numerals '3' and '5' were highly confused. This is illustrated in Table 6.7 showing the confusion matrix obtained by using the first 15 features from the ordering $D_A^2(2)$ and adopting the leave-one-out principle. Secondly, each of the two classes had 100 observations in it, thus making the sample size reasonably big for the analysis.

As in the case of 13-class experiments, all the redundant features were first discarded. 66 features were then left for further study. The orderings of these features were then obtained by the Bayesian error probability (P_e), the Bhattacharyya coefficient (ρ), and the divergence function (J) on the assumption of independence of features. An ordering was also obtained by the Mahalanobis distance criterion D_A^2 in two stages, first considering individual features and then, in stage II, rearranging the top 25 features taking into account their covariances. It may be noted here that since only one class-pair was dealt with, the results by D_A^2, D_B^2 and D^2 are the same. Similarly, ρ and γ give rise to the same ordering of features. Recognition experiments were then conducted, for feature subsets of various sizes, using the above four feature orderings, namely, $D_A^2(2)$, P_e, ρ and J. Leave-one-out principle was adopted in these experiments. Fig. 6.7 gives a diagrammatic representation of the recognition results. As in the case of 13-class problem, all the probabilistic criteria resulted in similar recognition accuracies. It is interesting to note that the best feature selected by all the criteria was the same which resulted in a recognition accuracy of 90.5%. Addition of further features had hardly any positive impact on the recognition score. Ignoring the small variations in recognition

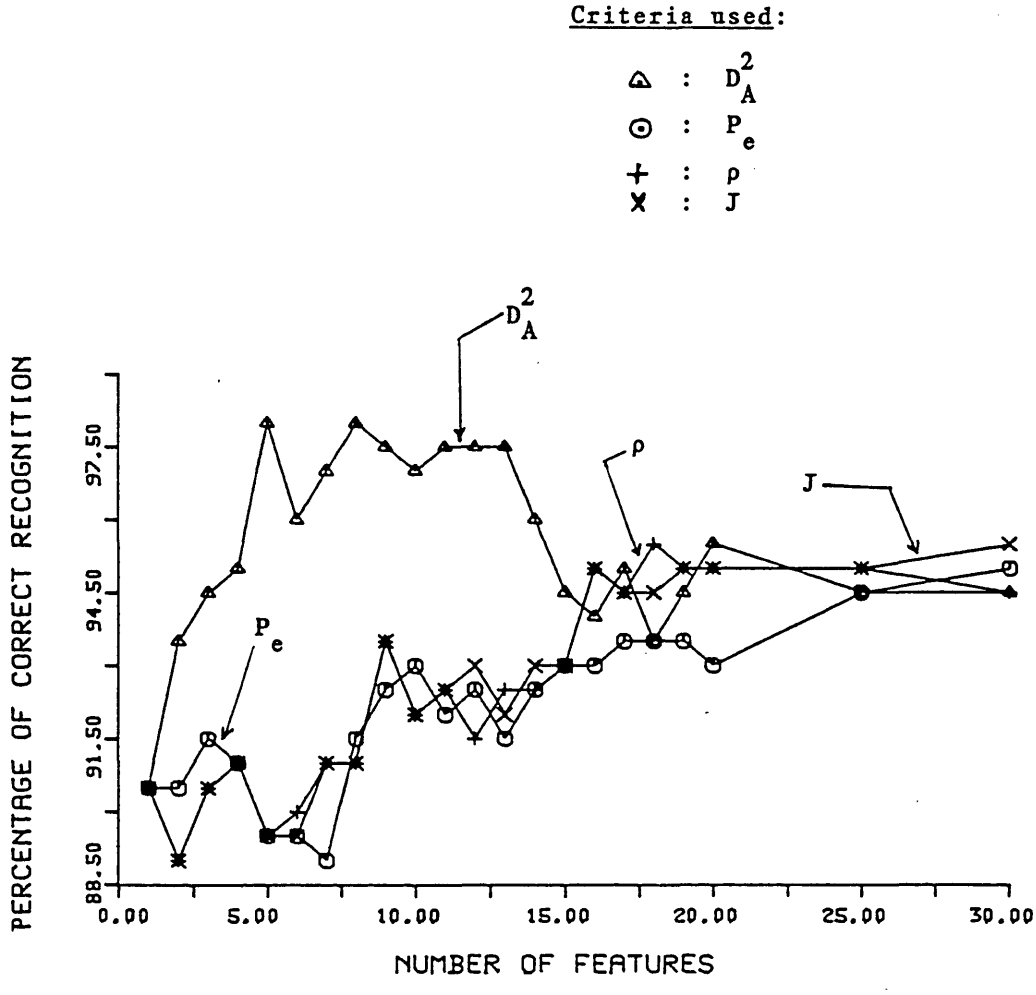


Fig. 6.7 Recognition accuracies in a 2-class situation (numerals '3' and '5'), by leave-one-out method, for features selected by D_A^2 (in two stages), P_e, ρ and J

Table 6.7
Confusion matrix obtained by using the first 15 features of the ordering $D_A^2(2)$ and adopting the leave-one-out principle

Actual numeral	Recognized as										Total
	0	1	2	3	4	5	6	7	8	9	
0	97	0	0	0	0	0	0	0	3	0	100
1	0	83	7	1	0	1	2	2	3	1	100
2	3	0	91	3	0	2	0	0	1	0	100
3	0	0	0	87	0	4	0	3	3	3	100
4	0	1	0	0	97	0	1	0	1	0	100
5	0	2	0	14	0	77	3	0	3	1	100
6	0	6	0	0	1	1	90	0	2	0	100
7	0	0	0	0	1	2	0	86	0	11	100
8	1	2	0	5	1	3	2	0	84	2	100
9	0	0	0	5	0	4	0	0	1	90	100

Correct recognition rate: 88.2%

scores one can say that, in the case of probabilistic criteria, with increase in feature subset size from 1 to 66 the recognition accuracy increased from 90.5% to 94.5%. In the case of the Mahalanobis distance the recognition accuracy rose to 98.0% for only 5 features. This shows the significance of the role played by the interactions of features in improving the recognition accuracy. Moreover, this result also reflects the usefulness of a simple feature evaluation criterion like the Mahalanobis distance. For higher number of features, however, the recognition accuracy decreased. This was probably due to the wellknown 'dimensionality problem', a topic not of interest as far as the present work is concerned.

6.5 Summary of Experimental Results

All the feature evaluation criteria led to quite similar recognition results. Recognition scores obtained by the leave-one-out principle were much less than those obtained by using the same data set for both training and testing. Depending on the feature evaluation criterion used and the number of features used, the difference between the two recognition scores was sometimes found to be more than 9%. Whether the same data were used for training and testing or the leave-one-out principle was adopted, the recognition results obtained by the use of features selected by different criteria were quite close to one another. In both cases P_e and $\bar{P}_{e2}$ produced slightly better results than the other probabilistic criteria. Though, in Stage I, the Mahalanobis distance-based criteria made use of only the means and the standard deviations of the features, the resulting recognition scores were comparable with those by the

probabilistic methods. Reordering of features in stage II by D_A^2 and D_B^2, taking into account the covariances of features, resulted in remarkable improvement in the recognition accuracy but D^2 failed to produce good results. This was in conformity with the speculation made in chapter 3 about the effectiveness of D_A^2 and D_B^2 compared to that of D^2.

At this point it is worthwhile to make some comments on the validity of the conclusions drawn on the basis of the recognition experiments. In all the recognition experiments the Bayesian classification criterion was used and, to avoid computational difficulties, the features were assumed to be independent. Though the classification criterion used was optimum from the point of view of recognition accuracy, the assumption of independence of features is unlikely to be true. Therefore the findings of the experiments cannot be taken as conclusive. There are two ways in which this situation could be improved. One way would be to work with a synthetic distribution such that the assumption of independence of features is assured. In this case the findings would be conclusive but only for the data which fit the distribution. The second alternative would be to take into account the interactions of features. But this is not an easy task. Estimation of higher order interactions would require a large-size data set. Another limitation involved in the present experiments was the use of features selected by a suboptimal procedure (see section 5.4). An exhaustive search would be needed to remove this limitation.

In the light of the above comments it is suggested that, though the experimental findings conform well with the conjectures made earlier about the effectiveness of various feature evaluation criteria including D_A^2 and D_B^2, more controlled experiments are required to reach decisive conclusions.

CHAPTER 7

CONCLUSIONS

7.1 Summary of Contributions

Criteria for the evaluation of effectiveness of features in pattern recognition were studied both theoretically and experimentally. Theoretical study consisted of (i) a comparative analysis of various probabilistic separability/distance measures from the point of view of their relationship with the Bayesian error probability (P_e) and (ii) the development of two Mahalanobis distance-based criteria. The experimental study consisted of applying some of the existing criteria and the two proposed criteria in the problem of recognition of isolated handprinted numerals.

Since, in many cases, the two-class measures form the basis for the development of multiclass measures, the two-class measures were discussed in some detail. In the consideration of two-class measures particular attention was given to the Bhattacharyya distance and the divergence function since they assume closed-form expressions in the main family of exponential distributions. It was shown that the maximum difference between the existing upper and (tighter) lower P_e bounds provided by the Bhattacharyya coefficient cannot exceed $\frac{1}{2}(\sqrt{2} - 1)$ whatever be the a'priori probabilities of the classes (see section 2.2.1).

From the analysis of the development of the probabilistic measures it was understood that the trend of research was towards defining new measures in order to tighten the existing error probability bounds or to generalize the existing measures. Though some success has been achieved from the point of view of the above two objectives, all the suggested measures appear to suffer from the same fundamental difficulties that are associated with the direct computation of P_e. It was found that the Bayesian distance has tighter error bounds than most of the other existing measures. But even with this measure, for m = 2, the difference between the upper and the lower bounds can be as large as 0.125 which, in practical pattern recognition problems, is likely to be too large (see section 2.3.4).

The Mahalanobis distance (Δ^2) is a simple measure defined in terms of the first and the second order moments only. Existence of a distribution-free upper bound on P_e in terms of Δ^2 has led to the present investigation of the usefulness of the Mahalanobis distance as a feature evaluation criterion. Two new Δ^2-based criteria, namely Δ_A^2 and Δ_B^2, were proposed for feature evaluation. They increase monotonically with increase in Δ^2. Δ^2 increases unboundedly but both Δ_A^2 and Δ_B^2 are upper bounded by 1. Because of this boundedness they are expected to perform better than Δ^2 in a multiclass pattern recognition problem (for details see section 3.3.1). It was conjectured that the two Mahalanobis distance-based F-statistics used for testing the difference between class means and for testing the sufficiency of a subset of features, respectively, may be applied for

evaluation of features.

Various probabilistic and Mahalanobis distance-based criteria have been applied for evaluation of features in the context of recognition of handprinted numerals. Though the application area of handprinted numeral recognition was chosen for the purpose of experimental comparison of various feature evaluation measures, the study was also a contribution to the character recognition research itself. A heuristic noise reduction scheme was developed for the removal of isolated noise specks from the character images. This scheme makes use of window sizes varying between 3 x 3 and 5 x 5. Uses of features like 'counts of '1'-valued pixels in subimages' and 'characteristic loci' are not new in character recognition. In this study the above features were normalized with a view to making them less susceptible to variations in character size.

Of the initial 101 featurs extracted from the characters 23 features were eliminated by a simple preliminary analysis which, in line with the terminology of Toussaint [80], may be called a 'pre-evaluation process'. The remaining 78 features were used in the feature ordering experiments.

For the selection of features on the basis of the Mahalanobis distance-based criteria a two-stage procedure was followed. In the first stage the features were evaluated individually, thus the computation of D^2 requiring information on only the means and the standard deviations of the features. The top 25 features obtained in the first stage were rearranged in the second stage by taking into

account their covariances and following a step-by-step procedure of inclusion of features starting with one feature. To get a better comparative picture of the three criteria, namely D_A^2, D_B^2 and D^2, they were applied, in stage II, on the same 25 features. The top 25 features selected by D_A^2 in stage I were used for this purpose. Computation of D^2 involves inversion of the covariance matrices. Likely computational problems, arising out of attempts to compute the inverse of a singular matrix, were avoided by considering the generalized inverse. In order to measure the conformity of a pair of feature orderings, Kendall's rank correlation coefficient was used instead of more widely used Spearman's rank correlation coefficient. This was on account of certain distributional as well as computational advantages of the former over the latter (see section 5.4.1).

Bayesian decision procedure, simplified under the assumption of independence of features, was used as the classification criterion in the recognition experiments. Recognition results were obtained by two approaches: (i) using the same training and test data, (ii) following the leave-one-out principle. In each of the two approaches an analysis of the recognition results obtained by the use of features selected by different feature evaluation criteria leads to a comparative assesment of the criteria. As far as the comparison of feature evaluation criteria is considered, the above two approches produced similar results. In addition, comparison of the two sets of results established the existence of severe overestimation of recognition accuracy in the case of the use of the same data for both training and testing.

In the second stage the selection of features was made from 25 features only. Moreover, in the step-by-step procedure followed for the selection of a subset of features all the possible combinations were not considered. Still the improvement gained in stage II was quite remarkable (see Figs. 6.1 and 6.5). In view of the fact that the implementation of a Mahalanobis distance-based criterion is much simpler than the implementation of a probabilistic criterion, the experimental results were very encouraging. Experimental results indicated the superiority of D_A^2 and D_B^2 over D^2 (see Fig. 6.4). This was in line with the conjecture made in section 3.3.1. In general, D_A^2 showed better performance than D_B^2. D_A^2 is derived on the basis of a distribution-free P_e upper bound whereas derivation of D_B^2 involves assumption of Gaussian distribution. Moreover, D_A^2 is computationally simpler than D_B^2. From all these D_A^2 is favoured over D_B^2 as a feature evaluation criterion.

7.2 Suggestions for Further Research

Further research aimed at defining new probabilistic criteria for the purposes of generalization of existing criteria and tightening of existing P_e bounds does not seem to be of much value as far as the solution of a practical pattern recognition problem is concerned. Instead, more emphasis should be given to finite sample estimation of the existing criteria. So far, in most of the real life applications of the probabilistic criteria either the features have been assumed to follow Gaussian distribution or, in the nonparametric cases, the features have been assumed to be independent. Nonparametric

estimation procedures are required in which higher order interactions can be taken into account.

In the present thesis emphasis was devoted to the development of simple feature evaluation criteria. Experimental results are quite encouraging. Effort is needed to derive theoretical properties of the proposed Mahalanobis distance-based criteria. Attempts should be made to develop similar criteria which would work directly for m classes.

The two F-statistics described in section 3.4 were not investigated experimentally. In a multiclass problem with varying sample sizes the degrees of freedom of these statistics would vary from class-pair to class-pair, thus making the application of a straightforward expected value approach unjustified. Experimental investigation of the two statistics should be made after arriving at a solution to the above mentioned difficulty.

In this thesis recognition experiments were conducted using the Bayesian classifier. Though the Bayesian classifier is an optimum classification criterion, the assumption of independence of features made in the experiments would have some adverse effect on the recognition scores. Therefore it would be interesting to perform the recognition experiments using classifiers such as K-Nearest Neighbour and minimum distance classifiers.

In the present study experimental comparison of various feature evaluation methods was made by applying them on a particular data set, namely, a set of numeric characters. It might be worth applying them

on different types of data to see if the comparative picture remained the same.

It was argued in this thesis that the 'normalization' process applied on the two sets of features extracted from the numeric characters should make the features less susceptible to variation in character size. Recognition experiments need to be conducted with 'unnormalized' features to examine the impact of the above normalization process.

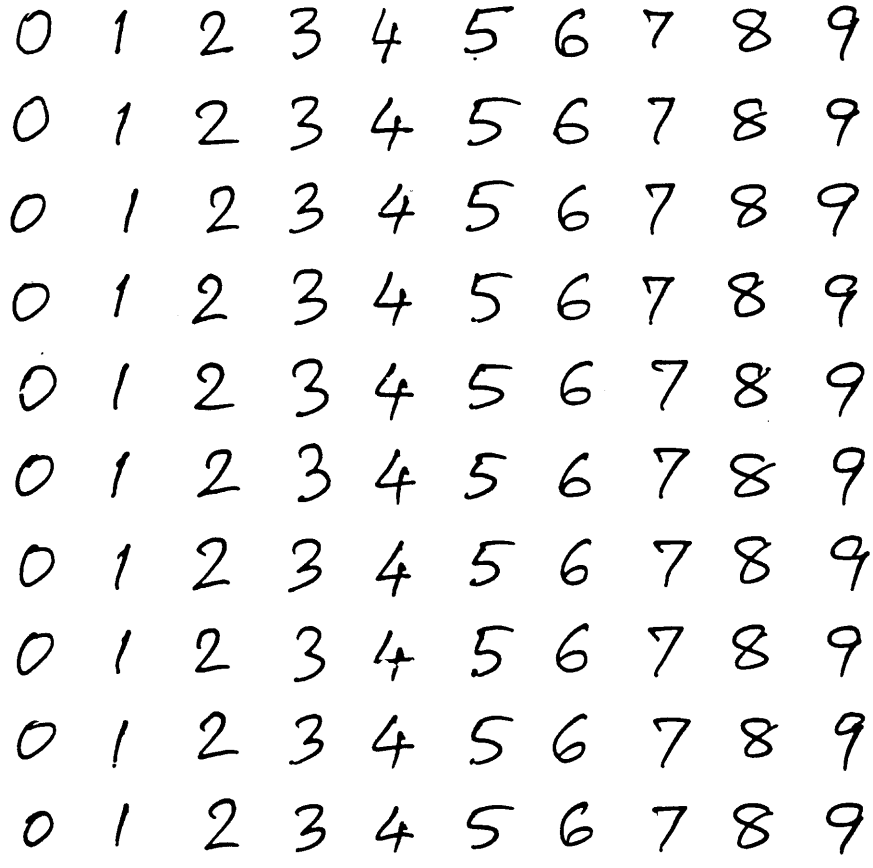
In this thesis experimental comparison of the feature evaluation criteria was made in a distribution-free situation. As pointed out in sections 5.5 and 6.5 (pages 152 and 181), experimental comparisons were carried out under conditions which are not likely to be valid in practice. In view of this, as suggested in the above sections, further experiments are necessary to reach decisive conclusions about the effectiveness of various feature evaluation criteria.

The main focus of the present study was to make a comparative assessment of feature evaluation criteria in a multiclass pattern recognition environment, whereas most of the investigated criteria are basically of two-class type. The relationship between the experimental values of a feature evaluation criterion and the error probability was not verified with the theoretical probability of error bounds. Two-class experiments should be conducted to study this important aspect.

In the present experiments statistical features were considered. Research should be pursued to extend the applicability of various feature evaluation criteria, more particularly the simple criteria like those derived based on the Mahalanobis distance, to syntactic features also. Success in this effort would lead to a much wider application area of the criteria.

APPENDIX A

1. Shiddhi



2. Mike

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

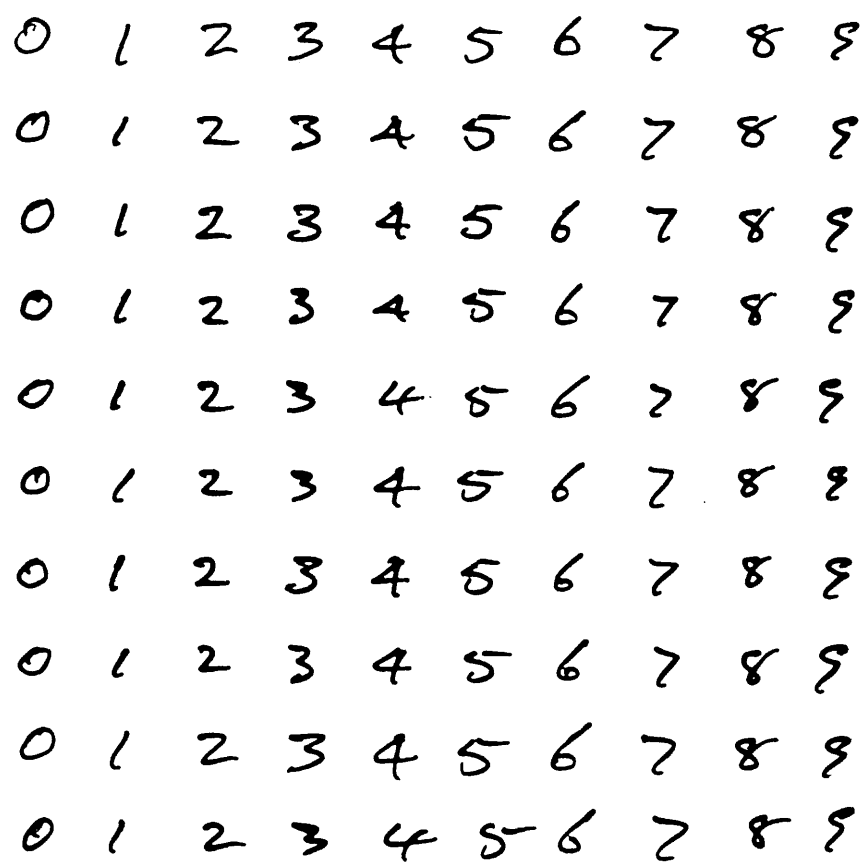
3. Tessa

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

4. Isam

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

5 Wilf



6. Sam

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

7. Jan

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

8. Tan

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

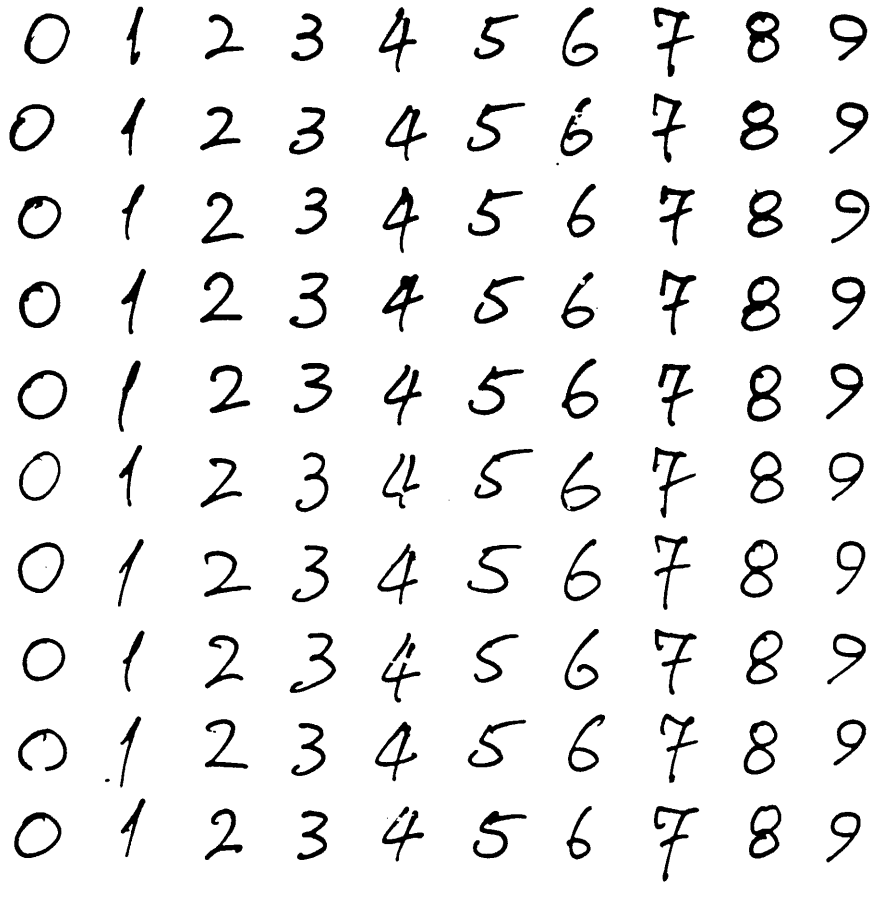
0 1 2 3 4 5 6 7 8 9

0 : 2 3 4 5 6 7 8 9

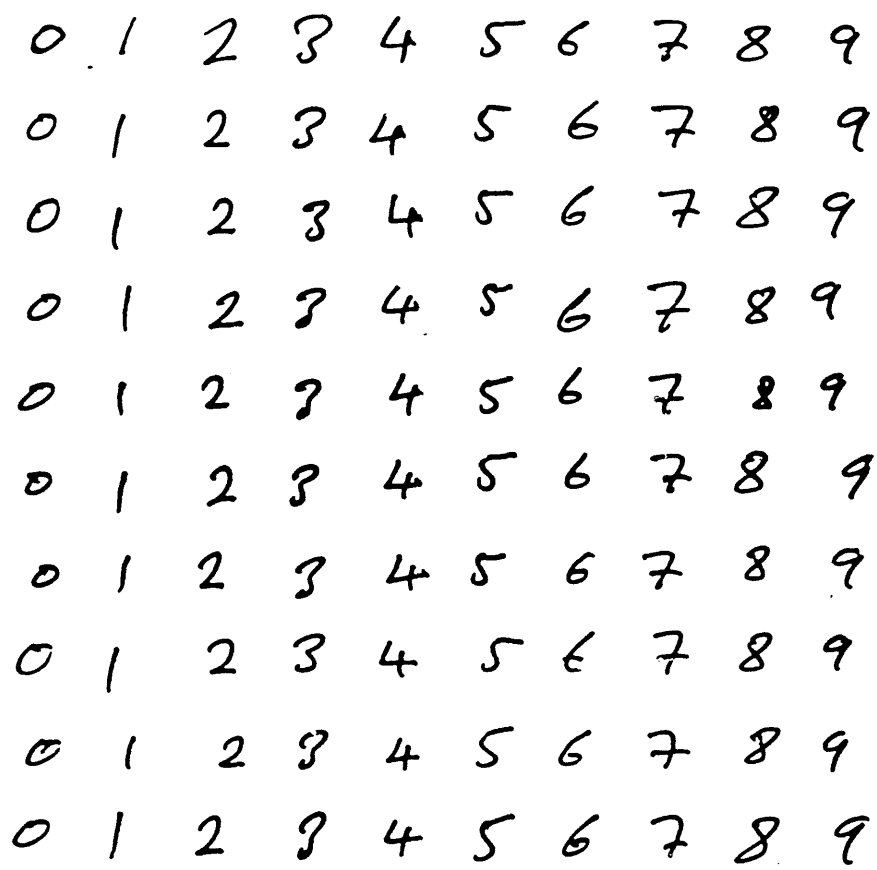
0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

9. Chen



10. Mazhar


0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

APPENDIX B1

```
C
C  UNPACK: PROGRAM FOR UNPACKING OF DIGITIZED IMAGES
C
      PROGRAM UNPACK(INPUT,OUTPUT,NOVA,D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,
+     TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=NOVA,TAPE11=D1,TAPE12=D2,
+     TAPE13=D3,TAPE14=D4,TAPE15=D5,TAPE16=D6,TAPE17=D7,TAPE18=D8,
+     TAPE19=D9,TAPE20=D10)
C
C  INPUT FILE:
C               NOVA: DIGITIZED IMAGE CONTAINING 100 NUMERAL SAMPLES
C  OUTPUT FILES:
C               D1, D2, ..., D10: UNPACKED VERSIONS OF 10 NUMERAL
C                                 SAMPLES OBTAINED FROM A SPECIFIED
C                                 FRAME (OUT OF 10 FRAMES)
C
      DIMENSION INOVA(612),INCDC(100)
C
C  IMAGE SIZE BEFORE UNPACKING (100 NUMERALS):
C
      IY=608
      IX=600
C
C  IMAGE SIZE AFTER UNPACKING (EACH NUMERAL):
C
      IYSIZE=60
      IXSIZE=60
C
C  PAIRS OF (AND NO. OF) CDC WORDS USED FOR UNPACKING:
C
      IPAIRS=(IY+4)/15+1
      NUM=INT(FLOAT(IY+4)/7.5)+1
C
C  SPECIFY THE LOCATION OF THE FRAME (CONTAINING 10 NUMERALS)
C  TO BE PROCESSED IN ONE RUN (TO UNPACK 100 NUMERALS WRITTEN IN
C  ONE PAGE THE PROGRAM IS EXECUTED 10 TIMES WITH FRAME LOCATIONS
C  1, 2, ..., 10):
C
      PRINT*,'FRAME LOCATION:INX'
      READ(5,*)INX
      IF(INX*IXSIZE.GT.IX)THEN
      PRINT*,'LOCATION BEYOND RANGE!'
      GO TO 500
      END IF
C
C  READ THE ABOVE SPECIFIED FRAME OF THE IMAGE IN MEMORY
C  AND DO THE UNPACKING
C
      NUNIT=7
      IF(INX.EQ.1)GOTO 90
      DO 10 I=1,(INX-1)*IXSIZE
```

```
      BUFFER IN(NUNIT,1),(INCDC(1),INCDC(NUM))
      IF(UNIT(NUNIT))10,30,40
40    PRINT*,'TAPE PARITY ERROR!'
      GO TO 500
30    PRINT*,'RECORD TOO SHORT!'
      GO TO 500
10    CONTINUE
90    IY1=IY+4
      DO 400 I=1,IXSIZE
C
C  CALL THE UNPACKING ROUTINE
C
      CALL TRANSL(NUNIT,INOVA,IY1,IPAIRS,NUM,INCDC)
C
C  WRITE THE DATA RECORDS RELATING TO 10 NUMERALS IN THE
C  10 FILES D1, D2, ..., D10.
C
      DO 399 IUNIT=11,20
      ILOW=(IUNIT-11)*IYSIZE+5
      IHIGH=ILOW+IYSIZE-1
      WRITE(IUNIT,250)(INOVA(K),K=ILOW,IHIGH)
399   CONTINUE
250   FORMAT(60Z2)
400   CONTINUE
500   STOP
      END
C
C   UNPACKING SUBROUTINE
C
      SUBROUTINE TRANSL(IUNIT,INOVA,IROW,IPAIRS,NUM,INCDC)
      DIMENSION INCDC(NUM),INOVA(IROW)
      BUFFER IN(IUNIT,1),(INCDC(1),INCDC(NUM))
      IF (UNIT(IUNIT))110,150,120
120   INOVA(1)=111
      RETURN
150   LEN=LENGTH(IUNIT)
      IF(LEN.EQ.NUM)GO TO 130
      INOVA(1)=333
      RETURN
110   LEN=LENGTH(IUNIT)
      IF(LEN.EQ.NUM)GO TO 130
      INOVA(1)=222
      RETURN
C
  130 IT=0
      NP=0
      DO 140 I=1,IPAIRS
      IT=IT+1
      INOVA(NP+1)=AND(SHIFT(INCDC(IT),-52),377B)
      INOVA(NP+2)=AND(SHIFT(INCDC(IT),-44),377B)
      INOVA(NP+3)=AND(SHIFT(INCDC(IT),-36),377B)
      INOVA(NP+4)=AND(SHIFT(INCDC(IT),-28),377B)
      INOVA(NP+5)=AND(SHIFT(INCDC(IT),-20),377B)
      INOVA(NP+6)=AND(SHIFT(INCDC(IT),-12),377B)
      INOVA(NP+7)=AND(SHIFT(INCDC(IT),-4), 377B)
```

```
      INOVA(NP+8)=OR(AND(SHIFT(INCDC(IT),4),360B),
     +                    AND(SHIFT(INCDC(IT+1),-56),17B))
     C
      IT=IT+1
      INOVA(NP+9)=AND(SHIFT(INCDC(IT),-48),377B)
      INOVA(NP+10)=AND(SHIFT(INCDC(IT),-40),377B)
      INOVA(NP+11)=AND(SHIFT(INCDC(IT),-32),377B)
      INOVA(NP+12)=AND(SHIFT(INCDC(IT),-24),377B)
      INOVA(NP+13)=AND(SHIFT(INCDC(IT),-16),377B)
      INOVA(NP+14)=AND(SHIFT(INCDC(IT),-8), 377B)
      INOVA(NP+15)=AND(INCDC(IT),377B)
      NP=NP+15
  140 CONTINUE
      RETURN
      END
```

APPENDIX B2

```
C
C  CONV: PROGRAM FOR GREY-TONE TO TWO-TONE CONVERSION OF CHARACTER
C        MATRICES (THIS PROGRAM CAN GET, PROCESS AND REPLACE A NO.
C        OF FILES, STORED IN PERMANENT FILE BASE, FROM WITHIN A
C        FORTRAN PROGRAM)
C
      PROGRAM CONV(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+      TAPE10,TAPE20,TAPE1)
C
C
C  INPUT FILE:
C                TAPE10: GREY-TONE IMAGE
C  OUTPUT FILES:
C                TAPE20: TWO-TONE IMAGE
C  WORKING FILES:
C                TAPE1:  USED TO GENERATE THE NAMES OF INPUT AND
C                        OUTPUT FILES
C
      DIMENSION IA(60,60)
      INTEGER CODE
      CHARACTER PFCMD1*80,MSG*40,CCODEI*2,CCODEO*2
      CHARACTER PFCMD2*80
      CHARACTER NMFLI*5,NMFLO*5,MRFILES*3
      WRITE(6,599)
  599 FORMAT(//1X,'ENTER INPUT MATRIX SIZE:NROWS,NCOLS')
      READ(5,*)NROWS,NCOLS
    2 WRITE(6,600)
  600 FORMAT(/1X,'ENTER NO. OF FILES TO BE PROCESSED?')
      READ(5,*)NFL
      WRITE(6,601)
  601 FORMAT(/1X,'ENTER CHARACTER CODE FOR INPUT FILES?')
      READ(5,500)CCODEI
  500 FORMAT(1A2)
      WRITE(6,604)
  604 FORMAT(/1X,'ENTER CHARACTER CODE FOR OUTPUT FILES?')
      READ(5,500)CCODEO
      WRITE(6,602)
  602 FORMAT(/1X,'ENTER NUMERIC PART OF 1ST FILE TO BE PROCESSED?')
      READ(5,*)N1F
      NPF=N1F-1
C
C  PROCESSING OF THE SPECIFIED NO. OF FILES STARTS HERE:
C
      DO 1 I=1,NFL
C
C  GENERATE THE NAME OF THE INPUT FILE AND WRITE IT ON TAPE1
C
      NPF=NPF+1
      REWIND 1
      IF(NPF.LE.9)THEN
```

```
      WRITE(1,100)CCODEI,NPF
  100 FORMAT(1A2,'00',I1)
      ELSE IF(NPF.GT.9.AND.NPF.LE.99)THEN
      WRITE(1,101)CCODEI,NPF
  101 FORMAT(1A2,'0',I2)
      ELSE IF(NPF.GT.99)THEN
      WRITE(1,102)CCODEI,NPF
  102 FORMAT(1A2,I3)
      ENDIF
      REWIND 1
      READ(1,103)NMFLI
  103 FORMAT(1A5)
      PFCMD2='RETURN,TAPE10.'
      CALL PFREQ(PFCMD2,MSG,CODE)
      PFCMD2='RETURN,TAPE20.'
      CALL PFREQ(PFCMD2,MSG,CODE)
C
C  OBTAIN, IN TAPE10, THE ABOVE INPUT FILE FROM THE PERMANENT
C  FILE BASE
C
      PFCMD1='GET,TAPE10='//NMFLI//'.'
      CALL PFREQ(PFCMD1,MSG,CODE)
      IF(CODE.NE.0)THEN
      PRINT*,MSG
      STOP
      ENDIF
      REWIND 10
C
C  CALL THE SUBROUTINE 'BINCON' WHICH DOES THE GREY-TONE TO
C  TWO-TONE CONVERSION AND THEN PUTS THE OUTPUT FILE IN TAPE20
C
      CALL BINCON(IA,NROWS,NCOLS,10,20,7)
C      TAPE20 CREATED!
C
C  GENERATE THE NAME OF THE OUTPUT FILE AND WRITE IT ON TAPE1
C
      REWIND 1
      IF(NPF.LE.9)THEN
      WRITE(1,100)CCODEO,NPF
      ELSE IF(NPF.GT.9.AND.NPF.LE.99)THEN
      WRITE(1,101)CCODEO,NPF
      ELSE IF(NPF.GT.99)THEN
      WRITE(1,102)CCODEO,NPF
      ENDIF
      REWIND 1
      READ(1,103)NMFLO
      REWIND 1
      REWIND 20
C
C  STORE THE OUTPUT FILE IN PERMANENT FILE BASE UNDER THE ABOVE NAME
C
      PFCMD1='REPLACE,TAPE20='//NMFLO//'.'
      CALL PFREQ(PFCMD1,MSG,CODE)
      IF(CODE.NE.0)THEN
      PRINT*,MSG
```

```
      STOP
      ENDIF
    1 CONTINUE
      WRITE(6,605)
  605 FORMAT(/1X,'REQUESTED NO. OF FILES HAVE BEEN PROCESSED'/
     +       1X,'DO YOU WISH TO PROCESS MORE FILES ?')
      READ(5,503)MRFILES
  503 FORMAT(1A3)
      IF(MRFILES.EQ.'YES')THEN
      GOTO 2
      ENDIF
      STOP
      END
C
      SUBROUTINE BINCON(IA,NROWS,NCOLS,TAPEIN,TAPEOUT,ITHRS)
C
      INTEGER TAPEIN,TAPEOUT
      DIMENSION IA(100)
      REWIND TAPEIN
      REWIND TAPEOUT
      DO 10 I=1,NROWS
      READ(TAPEIN,100)(IA(K),K=1,NCOLS)
  100 FORMAT(60Z2)
      DO 20 K=1,NCOLS
      IF(IA(K).LE.ITHRS)THEN
      IA(K)=0
      ELSE
      IA(K)=1
      END IF
   20 CONTINUE
      WRITE(TAPEOUT,200)(IA(K),K=1,NCOLS)
  200 FORMAT(1X,60I1)
   10 CONTINUE
      RETURN
      END
```

APPENDIX B3

```
C
C      MULTNS: THIS PROGRAM CLEANS NOISE FROM A SPECIFIED
C              NUMBER OF FILES
C
      PROGRAM MULTNS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+      TAPE10,TAPE20,TAPE1)
C
C  INPUT FILE:
C                TAPE10: BINARY IMAGE TO BE CLEANED
C  OUTPUT FILE:
C                TAPE20: BINARY IMAGE AFTER CLEANING OF NOISE
C  WORKING FILE:
C                TAPE1: USED TO GENERATE THE NAMES OF INPUT AND
C                       OUTPUT FILES
C
      DIMENSION IA(0:63,-1:62)
      INTEGER CODE
      CHARACTER PFCMD1*80,MSG*40,CCODEI*2,CCODEO*2
      CHARACTER PFCMD2*80
      CHARACTER NMFLI*5,NMFLO*5,MRFILES*3
      WRITE(6,599)
  599 FORMAT(//1X,'ENTER INPUT MATRIX SIZE:NROWS,NCOLS')
      READ(5,*)NROWS,NCOLS
    2 WRITE(6,600)
  600 FORMAT(/1X,'ENTER NO. OF FILES TO BE PROCESSED?')
      READ(5,*)NFL
      WRITE(6,601)
  601 FORMAT(/1X,'ENTER  CHARACTER  CODE FOR INPUT FILES?')
      READ(5,500)CCODEI
  500 FORMAT(1A2)
      WRITE(6,604)
  604 FORMAT(/1X,'ENTER CHARACTER CODE FOR OUTPUT FILES?')
      READ(5,500)CCODEO
      WRITE(6,602)
  602 FORMAT(/1X,'ENTER NUMERIC PART OF 1ST FILE TO BE PROCESSED?')
      READ(5,*)N1F
      NPF=N1F-1
      DO 1 II=1,NFL
C
C  GENERATE THE NAME OF THE INPUT FILE AND WRITE IT ON TAPE1
C
      NPF=NPF+1
      REWIND 1
      IF(NPF.LE.9)THEN
      WRITE(1,100)CCODEI,NPF
  100 FORMAT(1A2,'00',I1)
      ELSE IF(NPF.GT.9.AND.NPF.LE.99)THEN
      WRITE(1,101)CCODEI,NPF
  101 FORMAT(1A2,'0',I2)
      ELSE IF(NPF.GT.99)THEN
```

```
      WRITE(1,102)CCODEI,NPF
  102 FORMAT(1A2,I3)
      ENDIF
      REWIND 1
      READ(1,103)NMFLI
  103 FORMAT(1A5)
      PFCMD2='RETURN,TAPE10.'
      CALL PFREQ(PFCMD2,MSG,CODE)
      PFCMD2='RETURN,TAPE20.'
C
C  OBTAIN, IN TAPE10, THE ABOVE INPUT FILE FROM THE PERMANENT
C  FILE BASE
C
      CALL PFREQ(PFCMD2,MSG,CODE)
      PFCMD1='GET,TAPE10='//NMFLI//'.'
      CALL PFREQ(PFCMD1,MSG,CODE)
      IF(CODE.NE.0)THEN
      PRINT*,MSG
      STOP
      ENDIF
      REWIND 10
C
C      READ THE CONTENTS OF INPUT FILE (TAPE10) IN MATRIX IA
C
      DO 15 I=1,NROWS
      READ(10,200)(IA(I,J),J=1,NCOLS)
  200 FORMAT(1X,60I1)
   15 CONTINUE
      CALL NRSUB(IA,NROWS,NCOLS)
C
C      WRITE THE CONTENTS OF PROCESSED MATRIX ON OUTPUT FILE TAPE20
C
      REWIND 20
      DO 16 I=1,NROWS
      WRITE(20,200)(IA(I,J),J=1,NCOLS)
   16 CONTINUE
C      TAPE20 CREATED!
      REWIND 1
      IF(NPF.LE.9)THEN
      WRITE(1,100)CCODEO,NPF
      ELSE IF(NPF.GT.9.AND.NPF.LE.99)THEN
      WRITE(1,101)CCODEO,NPF
      ELSE IF(NPF.GT.99)THEN
      WRITE(1,102)CCODEO,NPF
      ENDIF
      REWIND 1
      READ(1,103)NMFLO
      REWIND 1
      REWIND 20
      PFCMD1='REPLACE,TAPE20='//NMFLO//'.'
      CALL PFREQ(PFCMD1,MSG,CODE)
      IF(CODE.NE.0)THEN
      PRINT*,MSG
      STOP
      ELSE
```

```
      WRITE(6,607)NMFLO
  607 FORMAT(1X,'OUTPUT FILE SAVED UNDER THE NAME',3X,,A5)
      ENDIF
    1 CONTINUE
      WRITE(6,605)
  605 FORMAT(/1X,'REQUESTED NO. OF FILES HAVE BEEN PROCESSED'/
     +       1X,'DO YOU WISH TO PROCESS MORE FILES ?')
      READ(5,503)MRFILES
  503 FORMAT(1A3)
      IF(MRFILES.EQ.'YES')THEN
      GOTO 2
      ENDIF
      STOP
      END
C
C  SUBROUTINE INCORPORATING THE NOISE REDUCTION ALGORITHM
C
      SUBROUTINE NRSUB(A,NROWS,NCOLS)
      INTEGER A,SUM
      DIMENSION A(0:NROWS+3,-1:NCOLS+2)
      DIMENSION SUM(20)
C
C      MATRIX A(NROWS,NCOLS) CONTAINS THE ORIGINAL MATRIX
C
C      ADD EXTRA ROWS AND COLS CONTAINING ZEROES WHICH WILL BE
C      USED BY THE NOISE REDUCTION ALGORITHM
C
      DO 10 J=-1,NCOLS+2
      A(0,J)=0
   10 CONTINUE
      DO 1 I=1,NROWS
      A(I,-1)=0
      A(I,0)=0
      A(I,NCOLS+1)=0
      A(I,NCOLS+2)=0
    1 CONTINUE
      DO 2 I=NROWS+1,NROWS+3
      DO 3 J=-1,NCOLS+2
      A(I,J)=0
    3 CONTINUE
    2 CONTINUE
C
C      APPLY NOISE REDUCTION ALGORITHM ON MATRIX A
C
      DO 5 I=1,NROWS
      DO 6 J=1,NCOLS
      IF(A(I,J) .EQ. 9)GOTO 6
C
C      CHECK FOR CONDITION 1:
C      SUM OF PREVIOUS AND NEXT ROWS (6 PTS) =0
C
      ISMPR=A(I-1,J-1)+A(I-1,J)+A(I-1,J+1)
      ISMNR=A(I+1,J-1)+A(I+1,J)+A(I+1,J+1)
      SUM(1)=ISMPR+ISMNR
      IF(SUM(1) .EQ. 0)GOTO 7
```

```
C
C       CHECK FOR CONDITION 2:
C       SUM OF PREVIOUS AND NEXT COLS (6 PTS) =0
C
      ISMPC=A(I-1,J-1)+A(I,J-1)+A(I+1,J-1)
      ISMNC=A(I-1,J+1)+A(I,J+1)+A(I+1,J+1)
      SUM(2)=ISMPC+ISMNC
      IF(SUM(2) .EQ. 0)GOTO 7
C
C       CHECK FOR CONDITION 3:
C       SUM OF SURROUNDING 8 PTS =0 OR 1
C
      SUM(3)=SUM(1)+A(I,J-1)+A(I,J+1)
      IF(SUM(3) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 4:
C       SUM OF I-1,I+1,J-1,J+2 (10 PTS) =0 OR 1
C
      ISM2NC=A(I-1,J+2)+A(I,J+2)+A(I+1,J+2)
      SUM(4)=SUM(3)-A(I,J+1)+ISM2NC
      IF(SUM(4) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 5:
C       SUM OF I-1,I+1,J-2,J+1 (10 PTS) =0 OR 1
C
      ISM2PC=A(I-1,J-2)+A(I,J-2)+A(I+1,J-2)
      SUM(5)=SUM(3)-A(I,J-1)+ISM2PC
      IF(SUM(5) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 6:
C        SUM OF I-1,I+2,J-1,J+1 (10 PTS) =0 OR 1
C
      ISM2NR=A(I+2,J-1)+A(I+2,J)+A(I+2,J+1)
      SUM(6)=SUM(3)-A(I+1,J)+ISM2NR
      IF(SUM(6) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 7:
C       SUM OF I-1,I+1,J-2,J+2 (12 PTS) =0 OR 1
C
      SUM(7)=ISMPR+ISMNR+ISM2PC+ISM2NC
      IF(SUM(7) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 8:
C       SUM OF I-1 I+2,J-1,J+2 (12 PTS) =0 OR 1
C
      SUM(8)=ISMPR+ISM2NR+ISMPC-A(I-1,J-1)+ISM2NC+A(I+2,J+2)
      IF(SUM(8) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 9:
C       SUM OF I-1,I+2,J-2,J+1 (12 PTS) =0 OR 1
C
      SUM(9)=ISMPR+ISM2NR+ISM2PC+ISMNC-A(I-1,J+1)+A(I+2,J-2)
      IF(SUM(9) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 10:
```

```
C       SUM OF I-1,I+3,J-1,J+1 (12 PTS) =0 OR 1
C
        ISM3NR=A(I+3,J-1)+A(I+3,J)+A(I+3,J+1)
        ITPC=A(I,J-1)+A(I+1,J-1)+A(1+2,J-1)
        ITNC=A(I,J+1)+A(I+1,J+1)+A(I+2,J+1)
        SUM(10)=ISMPR+ISM3NR+ITPC+ITNC
        IF(SUM(10) .LE. 1)GOTO 7
C
C       CHECK FOR CONDITION 11:
C       SUM OF I-1,I+2,J-2,J+2 (14 PTS) =0,1, OR 2
C
        SUM(11)=ISMPR+ISM2NR+ISM2PC+ISM2NC+A(I+2,J-2)+A(I+2,J+2)
        IF(SUM(11) .LE. 2)GOTO 7
C
C       CHECK FOR CONDITION 12:
C       SUM OF I-1,I+3,J-1,J+2 (14 PTS) =0,1, OR 2
C
        SUM(12)=ISMPR+ISM3NR+ITPC+ISM2NC+A(I+2,J+2)+A(I+3,J+2)
        IF(SUM(12) .LE. 2)GOTO 7
C
C       CHECK FOR CONDITION 13:
C       SUM OF I-1,I+3,J-2,J+1 (14 PTS) =0,1, OR 2
C
        SUM(13)=ISMPR+ISM3NR+ISM2PC+A(I+2,J-2)+A(I+3,J-2)+ITNC
        IF(SUM(13) .LE. 2)GOTO 7
C
C       CHECK FOR CONDITION 14:
C       SUM OF I-1,I+3,J-2,J+2 (16 PTS) =0,1, OR 2
C
        SUM(14)=ISMPR+ISM3NR+ISM2PC+ISM2NC+A(I+2,J-2)+A(I+3,J-2)+
+        A(I+2,J+2)+A(I+3,J+2)
        IF(SUM(14) .GT. 2)GOTO 6
    7   A(I,J)=0
    6   CONTINUE
    5   CONTINUE
        RETURN
        END
```

APPENDIX B4

```
C
C  FREQ: PROGRAM FOR EXTRACTION OF 'NORMALIZED FREQUENCY' FEATURES
C
      PROGRAM FREQ(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+                  TAPE10,OUTF,TAPE20=OUTF,TAPE1)
C
C  INPUT FILE:
C                TAPE10: TEMPORARY FILE TO STORE CHARACTER MATRICES,
C                        ONE AT A TIME, FROM WHICH FEATURES ARE TO
C                        BE EXTRACTED
C  OUTPUT FILE:  OUTF:   NORMALIZED FREQUENCY FEATURES OF THE INPUT
C                        CHARACTER MATRICES
C  WORKING FILE: TAPE1:  USED TO GENERATE THE NAMES OF THE INPUT FILES
C
      INTEGER A(-1:62,-1:62)
      DIMENSION IX(20),X(20)
      CHARACTER MRFILES*3,CCODEI*2,NMFLI*5
C
      WRITE(6,600)
  600 FORMAT(/1X,'ENTER NBI: NO. OF BLOCKS IN VERTICAL (I) DIRECTION?')
      READ(5,*)NBI
      WRITE(6,601)
  601 FORMAT(/1X,'ENTER NBJ: NO OF BLOCKS IN HORIZ. (J) DIRECTION?')
      READ(5,*)NBJ
      REWIND 20
    2 WRITE(6,602)
  602 FORMAT(/1X,'ENTER NO. OF FILES TO BE PROCESSED?')
      READ(5,*)NFL
      WRITE(6,603)
  603 FORMAT(/1X,'ENTER CHARACTER CODE FOR INPUT FILES?')
      READ(5,500)CCODEI
  500 FORMAT(A2)
      WRITE(6,604)
  604 FORMAT(/1X,'ENTER NUMERIC PART OF 1ST FILE TO BE PROCESSED?')
      READ(5,*)N1F
      NROWS=60
      NCOLS=60
      N=NBI*NBJ
      NPF=N1F-1
      DO 1 LL=1,NFL
      NPF=NPF+1
C
C  GET AN INPUT FILE IN TAPE10 FROM PERMANENT FILE BASE
C
      CALL GETINF(CCODEI,NPF,NMFLI)
C
C  READ THE INPUT FILE IN THE MATRIX A
C
      CALL RDMAT(A,NROWS,NCOLS)
C
```

```
C  ADD EXTRA ROWS AND COLUMNS TO MATRIX A
C
      CALL ADDRC(A,NROWS,NCOLS)
C
C  DETERMINE THE CHARACTER RECTANGLE
C
      CALL NUMRECT(A,NROWS,NCOLS,IMIN,IMAX,JMIN,JMAX,IL,JL,KTOT)
C
C  DEVELOP THE FETURES
C
      CALL DEVX(A,N,NBI,NBJ,KTOT,IL,JL,IMIN,IMAX,JMIN,JMAX,X,SUMX,
+           NROWS,NCOLS)
C
      WRITE(20,200)NMFLI,SUMX,(X(I),I=1,N)
  200 FORMAT(1X,A5,F10.4,4F10.5/4(16X,4F10.5/))
    1 CONTINUE
C
C
C
      WRITE(6,605)
  605 FORMAT(/1X,'REQUESTED NO. OF FILES HAVE BEEN PROCESSED'/
+      1X,'DO YOU WISH TO PROCESS MORE FILES?')
      READ(5,501)MRFILES
  501 FORMAT(A3)
      IF(MRFILES.EQ.'YES')THEN
      GOTO 2
      ELSE
      CLOSE(20)
      ENDIF
      STOP
      END
  SUBROUTINE RDMAT(A,M,N)
C
C      THIS SUBROUTINE READS A CHARACTER MATRIX IN THE MATRIX A
C
      INTEGER A
      DIMENSION A(-1:M+2,-1:N+2)
      DO 1 I=1,M
      READ(10,100)(A(I,J),J=1,N)
  100 FORMAT(1X,60I1)
    1 CONTINUE
      RETURN
      END
C
C
 SUBROUTINE ADDRC(A,NROWS,NCOLS)
C
C      THIS SUBROUTINE ADDS 4 EXTRA ROWS AND 4 EXTRA COLUMNS TO A MATRIX
C
      INTEGER A(-1:NROWS+2,-1:NCOLS+2)
      DO 1 J=-1,NCOLS+2
      A(-1,J)=0
      A(0,J)=0
      A(NROWS+1,J)=0
      A(NROWS+2,J)=0
```

```
    1 CONTINUE
      DO 2 I=1,NROWS
      A(I,-1)=0
      A(I,0)=0
      A(I,NCOLS+1)=0
      A(I,NCOLS+2)=0
    2 CONTINUE
      RETURN
      END
  SUBROUTINE NUMRECT(A,M,N,I1,I2,J1,J2,IL,JL,KTOT)
C
C      THIS SUBROUTINE OBTAINS THE TWO ROWS AND TWO COLS SURROUNDING
C      A NUMERAL
C
      INTEGER A,S
      DIMENSION A(-1:M+2,-1:N+2),S(60)
      KTOT=0
      DO 1 I=1,M
      S(I)=0
      DO 2 J=1,N
      S(I)=S(I)+A(I,J)
      KTOT=KTOT+A(I,J)
    2 CONTINUE
    1 CONTINUE
      CALL MNMX(S,M,I1,I2)
      DO 3 J=1,N
      S(J)=0
      DO 4 I=1,M
      S(J)=S(J)+A(I,J)
    4 CONTINUE
    3 CONTINUE
      CALL MNMX(S,N,J1,J2)
      IL=I2-I1+1
      JL=J2-J1+1
      RETURN
      END
C
C
 SUBROUTINE DEVX(A,N,NBI,NBJ,KTOT,IL,JL,IMIN,IMAX,JMIN,JMAX,X,SUMX,
+            NROWS,NCOLS)
C
C  THIS SUBROUTINE DEVELOPS THE NORMALIZED FREQUENCY FEATURES
C
      INTEGER A(-1:NROWS+2,-1:NCOLS+2)
      DIMENSION X(20),IX(20)
C
C  SUBDIVIDE THE CHARACTER RECTANGLE INTO SMALL RECTANGULAR BOXES
C
      NVPBI=IL/NBI
      IREM=IL-NVPBI*NBI
      IF(IREM.GT.0)THEN
      NVPBI=NVPBI+1
      ENDIF
      NVPBJ=JL/NBJ
      JREM=JL-NVPBJ*NBJ
```

```
      IF(JREM.GT.0)THEN
      NVPBJ=NVPBJ+1
      ENDIF
C
C  DEPENDING ON THE NO. OF ROWS PER BOX, ADD EXTRA ROWS TO THE
C  CHARACTER RECTANGLE
C
      IF(IREM.EQ.0)THEN
      ISTRT=IMIN
      IEND=IMAX
      ELSEIF(IREM.EQ.1)THEN
      ISTRT=IMIN-2
      IEND=IMAX+2
      ELSEIF(IREM.EQ.2)THEN
      ISTRT=IMIN-2
      IEND=IMAX+1
      ELSEIF(IREM.EQ.3)THEN
      ISTRT=IMIN-1
      IEND=IMAX+1
      ELSEIF(IREM.EQ.4)THEN
      ISTRT=IMIN-1
      IEND=IMAX
      ENDIF
C
C  DEPENDING ON THE NO. OF COLUMNS PER BOX, ADD EXTRA COLUMNS TO THE
C  CHARACTER RECTANGLE
C
      IF(JREM.EQ.0)THEN
      JSTRT=JMIN
      JEND=JMAX
      ELSEIF(JREM.EQ.1)THEN
      JSTRT=JMIN-2
      JEND=JMAX+1
      ELSEIF(JREM.EQ.2)THEN
      JSTRT=JMIN-1
      JEND=JMAX+1
      ELSEIF(JREM.EQ.3)THEN
      JSTRT=JMIN-1
      JEND=JMAX
      ENDIF
C
C  DETERMINE THE FREQUENCIES OF '1'S IN THE BOXES
C
      DO 2 I=1,NBI
      DO 3 J=1,NBJ
      K=(I-1)*NBJ+J
      IX(K)=0
      DO 4 I1=ISTRT+(I-1)*NVPBI,ISTRT+I*NVPBI-1
      DO 5 J1=JSTRT+(J-1)*NVPBJ,JSTRT+J*NVPBJ-1
      IX(K)=IX(K)+A(I1,J1)
    5 CONTINUE
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
C
```

```
C  DETERMINE THE NORMALIZED FREQUENCY FEATURES
C
      SUMX=0
      DO 6 K=1,N
      X(K)=REAL(IX(K))/REAL(KTOT)
      SUMX=SUMX+X(K)
    6 CONTINUE
      RETURN
      END
C
C
 SUBROUTINE GETINF(CCODEI,NPF,NMFLI)
C
C  THIS SUBROUTINE GETS AN INPUT FILE IN TAPE10
C
      INTEGER CODE
      CHARACTER PFCMD1*80,PFCMD2*80,MSG*40
      CHARACTER CCODEI*2,NMFLI*5
      REWIND 1
      IF(NPF.LE.9)THEN
      WRITE(1,100)CCODEI,NPF
  100 FORMAT(1A2,'00',I1)
      ELSE IF(NPF.GT.9.AND.NPF.LE.99)THEN
      WRITE(1,101)CCODEI,NPF
  101 FORMAT(1A2,'0',I2)
      ELSE IF(NPF.GT.99)THEN
      WRITE(1,102)CCODEI,NPF
  102 FORMAT(1A2,I3)
      ENDIF
      REWIND 1
      READ(1,103)NMFLI
  103 FORMAT(1A5)
      PFCMD2='RETURN,TAPE10.'
      CALL PFREQ(PFCMD2,MSG,CODE)
      PFCMD1='GET,TAPE10='//NMFLI//'.'
      CALL PFREQ(PFCMD1,MSG,CODE)
      IF(CODE.NE.0)THEN
      PRINT*,MSG
      STOP
      ENDIF
      REWIND 10
      RETURN
      END
  SUBROUTINE MNMX(S,N,MIN,MAX)
C
C      GIVEN A STRING OF NUMBERS THIS SUBROUTINE OBTAINS THE
C      MINIMUM AND MAXIMUM LOCATIONS HAVING NONZERO VALUES
C
      INTEGER S(60)
      MIN=1
      MAX=N
      DO 1 I=2,N-1
      IF(S(I).GT.0)THEN
      IF(S(I-1).EQ.0)THEN
      MIN=I
```

```
      ENDIF
      IF(S(I+1).EQ.0)THEN
      MAX=I
      ENDIF
      ENDIF
    1 CONTINUE
      RETURN
      END
```

APPENDIX B5

```
C
C  CLOCI: PROGRAM FOR EXTRACTION OF NORMALIZED CHARACTERISTIC LOCI
C         FEATURES
C
      PROGRAM CLOCI(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+                   TAPE10,OUTF,TAPE20=OUTF,TAPE1)
C
C  INPUT FILE:
C                TAPE10: TEMPORARY FILE TO STORE CHARACTER MATRICES,
C                        ONE AT A TIME, FROM WHICH FEATURES ARE TO
C                        BE ESTIMATED
C  OUTPUT FILE:
C                OUTF:   NORMALIZED CHARACTERISTIC LOCI FEATURES OF
C                        THE INPUT CHARACTER MATRICES
C  WORKING FILE: TAPE1:  USED TO GENERATE THE NAMES OF THE INPUT FILES
C
      INTEGER A(-1:62,-1:62)
      DIMENSION IFRQ(0:80),X(0:80)
      CHARACTER MRFILES*3,CCODEI*2,NMFLI*5
C
      REWIND 20
    2 WRITE(6,602)
  602 FORMAT(/1X,'ENTER NO. OF FILES TO BE PROCESSED?')
      READ(5,*)NFL
      WRITE(6,603)
  603 FORMAT(/1X,'ENTER CHARACTER CODE FOR INPUT FILES?')
      READ(5,500)CCODEI
  500 FORMAT(A2)
      WRITE(6,604)
  604 FORMAT(/1X,'ENTER NUMERIC PART OF 1ST FILE TO BE PROCESSED?')
      READ(5,*)N1F
      NROWS=60
      NCOLS=60
      NPF=N1F-1
      DO 1 LL=1,NFL
      NPF=NPF+1
C
C  GET AN INPUT FILE IN TAPE10 FROM PERMANENT FILE BASE
C
      CALL GETINF(CCODEI,NPF,NMFLI)
C
C  READ THE INPUT FILE IN MATRIX A
C
      CALL RDMAT(A,NROWS,NCOLS)
C
C  DETERMINE THE CHARACTER RECTANGLE
C
      CALL NUMRECT(A,NROWS,NCOLS,IMIN,IMAX,JMIN,JMAX,IL,JL,KTOT)
C
C  DEVELOP THE FEATURES
```

```
C
      CALL LCROSS(A,IMIN,IMAX,JMIN,JMAX,IFRQ,ITFRQ,NMFLI)
      TFRQ=ITFRQ
      DO 3 I=0,80
      X(I)=REAL(IFRQ(I))/TFRQ
    3 CONTINUE
      WRITE(20,200)NMFLI,ITFRQ,(X(K),K=0,80)
  200 FORMAT(1X,A5,5X,I10/(1X,9F8.4))
    1 CONTINUE
      WRITE(6,605)
  605 FORMAT(/1X,'REQUESTED NO. OF FILES HAVE BEEN PROCESSED'/
+      1X,'DO YOU WISH TO PROCESS MORE FILES?')
      READ(5,501)MRFILES
  501 FORMAT(A3)
      IF(MRFILES.EQ.'YES')THEN
      GOTO 2
      ELSE
      CLOSE(20)
      ENDIF
      STOP
      END
C
C
C
 SUBROUTINE GETINF(CCODEI,NPF,NMFLI)
C
C  THIS SUBROUTINE GETS AN INPUT FILE IN TAPE10
C
      INTEGER CODE
      CHARACTER PFCMD1*80,PFCMD2*80,MSG*40
      CHARACTER CCODEI*2,NMFLI*5
      REWIND 1
      IF(NPF.LE.9)THEN
      WRITE(1,100)CCODEI,NPF
  100 FORMAT(1A2,'00',I1)
      ELSE IF(NPF.GT.9.AND.NPF.LE.99)THEN
      WRITE(1,101)CCODEI,NPF
  101 FORMAT(1A2,'0',I2)
      ELSE IF(NPF.GT.99)THEN
      WRITE(1,102)CCODEI,NPF
  102 FORMAT(1A2,I3)
      ENDIF
      REWIND 1
      READ(1,103)NMFLI
  103 FORMAT(1A5)
      PFCMD2='RETURN,TAPE10.'
      CALL PFREQ(PFCMD2,MSG,CODE)
      PFCMD1='GET,TAPE10='//NMFLI//'.'
      CALL PFREQ(PFCMD1,MSG,CODE)
      IF(CODE.NE.0)THEN
      PRINT*,MSG
      STOP
      ENDIF
      REWIND 10
      RETURN
```

```
      END
  SUBROUTINE RDMAT(A,M,N)
C
C      THIS SUBROUTINE READS THE CHARACTER MATRIX IN A
C
      INTEGER A
      DIMENSION A(-1:M+2,-1:N+2)
      DO 1 I=1,M
      READ(10,100)(A(I,J),J=1,N)
  100 FORMAT(1X,60I1)
    1 CONTINUE
      RETURN
      END
  SUBROUTINE NUMRECT(A,M,N,I1,I2,J1,J2,IL,JL,KTOT)
C
C      THIS SUBROUTINE OBTAINS THE TWO ROWS AND TWO COLS SURROUNDING
C      A NUMERAL
C
      INTEGER A,S
      DIMENSION A(-1:62,-1:62),S(60)
      KTOT=0
      DO 1 I=1,M
      S(I)=0
      DO 2 J=1,N
      S(I)=S(I)+A(I,J)
      KTOT=KTOT+A(I,J)
    2 CONTINUE
    1 CONTINUE
      CALL MNMX(S,M,I1,I2)
      DO 3 J=1,N
      S(J)=0
      DO 4 I=1,M
      S(J)=S(J)+A(I,J)
    4 CONTINUE
    3 CONTINUE
      CALL MNMX(S,N,J1,J2)
      IL=I2-I1+1
      JL=J2-J1+1
      RETURN
      END
C
C
  SUBROUTINE LCROSS(A,IMIN,IMAX,JMIN,JMAX,IFRQ,ITFRQ,NMFLI)
C
C  THIS SUBROUTINE DEVELOPS THE NORMALIZED CHARACTERISTIC LOCI
C  FEATURES
C
      INTEGER A(-1:62,-1:62)
      CHARACTER NMFLI*5
      DIMENSION IFRQ(0:80)
C
C      INITIALIZATION OF FREQUENCIES
C
      NZEROS=0
      DO 50 K=0,80
```

```
      IFRQ(K)=0
   50 CONTINUE
C
C       COMPUTATIONS
C
      DO 1 I=IMIN,IMAX
      DO 2 J=JMIN,JMAX
C
C
      IF(A(I,J).EQ.0)THEN
      NZEROS=NZEROS+1
C
C       COMPUTE THE FOUR LINE CROSSINGS
C
C       COMPUTE NCL: NO. OF CROSSINGS TOWARDS LEFT
C
      NCL=0
      IF(J.GT.JMIN)THEN
      DO 3 J1=J,JMIN+1,-1
      IF(A(I,J1).EQ.0.AND.A(I,J1-1).EQ.1)THEN
      NCL=NCL+1
      ENDIF
    3 CONTINUE
C
C       FOR NCL GREATER THAN 2 MAKE IT EQUAL TO 2
C
      IF(NCL.GT.2)THEN
      NCL=2
      ENDIF
      ENDIF
C
C       COMPUTE NCT: NO. OF CROSSINGS TOWARDS TOP
      NCT=0
      JF(I.GT.IMIN)THEN
      DO 6 I1=I,IMIN+1,-1
      IF(A(I1,J).EQ.0.AND.A(I1-1,J).EQ.1)THEN
      NCT=NCT+1
      ENDIF
    6 CONTINUE
      IF(NCT.GT.2)THEN
      NCT=2
      ENDIF
      ENDIF
C
C       COMPUTE NCR: NO. OF CROSSINGS TOWARDS RIGHT
C
      NCR=0
      IF(J.LT.JMAX)THEN
      DO 8 J1=J,JMAX-1
      IF(A(I,J1).EQ.0.AND.A(I,J1+1).EQ.1)THEN
      NCR=NCR+1
      ENDIF
    8 CONTINUE
      IF(NCR.GT.2)THEN
      NCR=2
```

```
      ENDIF
      ENDIF
C
C       COMPUTE NCB: NO. OF CROSSINGS TOWARDS BOTTOM
C
      NCB=0
      IF(I.LT.IMAX)THEN
      DO 10 I1=I,IMAX-1
      IF(A(I1,J).EQ.0.AND.A(I1+1,J).EQ.1)THEN
      NCB=NCB+1
      ENDIF
   10 CONTINUE
      IF(NCB.GT.2)THEN
      NCB=2
      ENDIF
      ENDIF
C
C       COMPUTE FREQUENCIES OF CODES 0,1,2, ...,80
C
      KVAL=27*NCL+9*NCT+3*NCR+NCB
      IFRQ(KVAL)=IFRQ(KVAL)+1
      ENDIF
    2 CONTINUE
    1 CONTINUE
C
C       CALCULATE ITFRQ: TOTAL FREQUENCY
C
      ITFRQ=0
      DO 51 K=0,80
      ITFRQ=ITFRQ+IFRQ(K)
   51 CONTINUE
C
C       CROSS-CHECK TOTAL FREQUENCY
C
      IF(NZEROS.NE.ITFRQ)THEN
      WRITE(6,601)NMFLI
  601 FORMAT(/1X,'FILE ',A5,': CALCULATION DOUBTFUL')
      STOP
      ENDIF
      RETURN
      END
  SUBROUTINE MNMX(S,N,MIN,MAX)
C
C       GIVEN A STRING OF NUMBERS THIS SUBROUTINE OBTAINS THE
C       MINIMUM AND MAXIMUM LOCATIONS HAVING NONZERO VALUES
C
      INTEGER S(60)
      MIN=1
      MAX=N
      DO 1 I=2,N-1
      IF(S(I).GT.0)THEN
      IF(S(I-1).EQ.0)THEN
      MIN=I
      ENDIF
      IF(S(I+1).EQ.0)THEN
```

```
      MAX=I
      ENDIF
      ENDIF
    1 CONTINUE
      RETURN
      END
```

APPENDIX B6

```
C
C  HISPDF: PROGRAM FOR ESIMATION OF CLASS-CONDITIONAL PROBABILITIES BY
C          HISTOGRAM APPROACH
C
C
      PROGRAM HISPDF(INPUT,OUTPUT,INF1,INF2,OUTF,TAPE5=INPUT,
     +          TAPE6=OUTPUT,TAPE10=INF1,TAPE11=INF2,TAPE20=OUTF)
C
C  INPUT FILES:
C                 INF1: FEATURE VALUES OF DIFFERENT CHARACTER SAMPLES
C                       BELONGING TO A NUMERAL CLASS
C                 INF2: MINIMUM AND MAXIMUM VALUES OF FEATURES
C
C  OUTPUT FILE:
C                 OUTF: ESTIMATED VALUES OF THE CLASS-CONDITIONAL
C                       PROBABILITIES
C
C
      DIMENSION X(100),XMIN(100),XMAX(100),FRQ(100,15)
      DIMENSION XSAMPL(100,20),H(100),XRANGE(100)
C
      WRITE(6,600)
  600 FORMAT(1X,'ENTER NO. OF FEATURES: N ?')
      READ(5,*)N
      WRITE(6,601)
  601 FORMAT(1X,'ENTER NO. OF OBSERVATIONS: NOBS ?')
      READ(5,*)NOBS
      WRITE(6,607)
  607 FORMAT(1X,'ENTER NO. OF OBS. TO SKIP: NSKIP ?')
      READ(5,*)NSKIP
      WRITE(6,602)
  602 FORMAT(1X,'ENTER NO. OF SAMPLING POINTS: NPNTS ?')
      READ(5,*)NPNTS
      REWIND 10
      REWIND 11
      REWIND 20
C
C READ THE MINIMUM AND MAXIMUM VALUES OF FEATURES
C
      READ(11,110)(XMIN(I),XMAX(I),I=1,N)
  110 FORMAT(1X,8F8.4)
C
C COMPUTE RANGES OF FEATURES
C
      ISWITCH=1
      DO 11 I=1,N
      XRANGE(I)=XMAX(I)-XMIN(I)
      IF(XRANGE(I).LT.1.0E-10)THEN
      ISWITCH=0
      WRITE(6,603)I,XRANGE(I)
```

```
  603 FORMAT(1X,'FEATURE NO. ',I4,'  HAS VERY SMALL RANGE:',E14.7)
      ENDIF
   11 CONTINUE
      IF(ISWITCH.EQ.0)THEN
      WRITE(6,604)
  604 FORMAT(1X,'DISCARD THE ABOVE FEATURES FROM INPUT LIST')
      STOP
      ELSE
C
C COMPUTE (INTERVAL LENGTH)/2 FOR THE FEATURES
C
      XNP=NPNTS
      DO 1 I=1,N
      H(I)=XRANGE(I)/(XNP*2.0)
    1 CONTINUE
C
C COMPUTE POINTS REPRESENTING SAMPLING INTERVALS
C
      DO 2 I=1,N
      DO 3 J=1,NPNTS
      FRQ(I,J)=0
      XSAMPL(I,J)=XMIN(I)+H(I)*(2.0*REAL(J)-1.0)
    3 CONTINUE
    2 CONTINUE
C
C SKIP DESIRED NO. OF OBSERVATIONS
C
      IF(NSKIP.GT.0)THEN
      DO 13 L=1,NSKIP
      READ(10,101)
      READ(10,102)(X(I),I=1,N)
   13 CONTINUE
      ENDIF
C
C DETERMINE NO. OF OBS. FALLING IN DIFFERENT SAMPLING INTERVALS
C
      DO 4 L=1,NOBS
      READ(10,101)
  101 FORMAT( )
      READ(10,102)(X(I),I=1,N)
  102 FORMAT(1X,8F8.4)
      DO 5 I=1,N
      IF(X(I).LT.(XSAMPL(I,1)+H(I)))THEN
      FRQ(I,1)=FRQ(I,1)+1.0
      GOTO 5
      ELSE IF(X(I).GE.(XSAMPL(I,NPNTS)-H(I)))THEN
      FRQ(I,NPNTS)=FRQ(I,NPNTS)+1.0
      GOTO 5
      ELSE
      DO 6 J=2,NPNTS-1
      IF((X(I).GE.(XSAMPL(I,J)-H(I))).AND.(X(I).LT.(XSAMPL(I,J)
+          +H(I))))THEN
      FRQ(I,J)=FRQ(I,J)+1.0
      GOTO 5
      ENDIF
```

```
    6 CONTINUE
      ENDIF
    5 CONTINUE
    4 CONTINUE
C
C ESTIMATE PROBS. OF SAMPLING INTERVALS
C
      XNOBS=NOBS
      DO 7 I=1,N
      DO 8 J=1,NPNTS
      FRQ(I,J)=FRQ(I,J)/XNOBS
    8 CONTINUE
    7 CONTINUE
C
C WRITE THE ESTMATED PROBS. IN OUTPUT FILE:OUTF
C
      DO 9 I=1,N
      WRITE(20,201)(FRQ(I,J),J=1,NPNTS)
  201 FORMAT(1X,5F8.4)
    9 CONTINUE
      ENDIF
      STOP
      END
```

APPENDIX B7

```
C
C  AVDIST:
C  FOR EACH OF A NO. OF FEATURES THIS PROGRAM CALCULATES THE VALUES
C  TAKEN BY A SPECIFIED PROBABILISTIC DISTANCE CRITERION BETWEEN
C  DIFFERENT CLASS-PAIRS,ITS AVERAGE OVER ALL CLASS-PAIRS, AND THEN
C  ARRANGES THE FEATURES IN DESCENDING/ASCENDING ORDER OF THE VALUES
C
      PROGRAM AVDIST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,INF,OUTF1,
+                    OUTF2,TAPE10=INF,TAPE21=OUTF1,TAPE22=OUTF2)
C
C NOTE: APRIORI PROBABILITIES ARE TAKEN TO BE EQUAL
C
C  INPUT FILE:
C                INF: CLASS-CONDITIONAL PROBABILITIES OF DIFFERENT
C                     FEATURES
C
C  OUTPUT FILES:
C                OUTF1: VALUES, BETWEEN VARIOUS CLASS-PAIRS AND FOR
C                       EACH FEATURE, THE VALUES OF THE SPECIFIED
C                       CRITERION
C
C                OUTF2: FEATURES ARRANGED IN DECREASING ORDER OF THEIR
C                       EFFECTIVENESS (TOGETHER WITH THE CRITERION-VALUES)
C
      DIMENSION P(13,10),D(13,13),AVD(78),AVDW(78),
+               ILOC(78),ILOCW(78)
      INTEGER U
      REAL KOLMS,MATUS
      CHARACTER IGNORE*3
      DATA MAXM/13/,MAXPNTS/10/
C
      REWIND 10
      REWIND 21
      REWIND 22
      WRITE(6,605)
  605 FORMAT(1X,'REFER TO THE FOLLOWING LIST TO SELECT DISTANCE'/
+      1X,'CRITERION IDENTIFICATION CODE:')
      WRITE(6,606)
  606 FORMAT(/12X,'BAYESIAN PROB. OF ERROR:          1'/
+             12X,'KOLMOGOROV VARIATIONAL DIST.:     2'/
+             12X,'BHATTACHARYYA COEFFICIENT:        3'/
+             12X,'MATUSITA DISTANCE:                4'/
+             12X,'DIVERGENCE FUNCTION:              5')
C
C  SPECIFY THE FEATURE EVALUATION CRITERION TO BE USED
C
      WRITE(6,651)
  651 FORMAT(//1X,'ENTER DIST. CRITERION IDENT. CODE: ICRIT')
      READ(5,*)ICRIT
      GOTO(11,12,13,14,15),ICRIT
```

```
   11 WRITE(21,215)
  215 FORMAT(///1X,'BAYESIAN ERROR PROBS FOR DIFFERENT CLASS-PAIRS'/
     +      1X,'WITH ASSUMPTION OF EQUAL APRIORI PROBS.')
      WRITE(22,225)
  225 FORMAT(///1X,'FEATURE ORDERING BY 2-CLASS BAYESIAN ERROR'/1X,
     +      'PROBABILITY METHOD, WITH EQUAL APRIORI PROBS.')
      GOTO 20
   12 WRITE(21,216)
  216 FORMAT(///1X,'KOLM. VAR. DISTANCES FOR DIFFERENT CLASS-PAIRS'/
     +      1X,'WITH ASSUMPTION OF EQUAL APRIORI PROBS.')
      WRITE(22,226)
  226 FORMAT(///1X,'FEATURE ORDERING BY KOLM. VAR. DIST.,'/1X,
     +       'WITH EQUAL APRIORI PROBABILITIES')
      GOTO 20
   13 WRITE(21,217)
  217 FORMAT(///1X,'BHATTACHARYYA COEFFS FOR DIFFERENT CLASS-PAIRS'/
     +      1X,'WITH ASSUMPTION OF EQUAL APRIORI PROBS.')
      WRITE(22,227)
  227 FORMAT(///1X,'FEATURE ORDERING BY BHATTACHARYYA COEFFS.,'/1X,
     +      'WITH EQUAL APRIORI PROBABILITIES')
      GOTO 20
   14 WRITE(21,218)
  218 FORMAT(///1X,'MATUSITA DISTANCES FOR DIFFERENT CLASS-PAIRS'/
     +      1X,'WITH ASSUMPTION OF EQUAL APRIORI PROBS.')
      WRITE(22,228)
  228 FORMAT(///1X,'FEATURE ORDERING BY MATUSITA DISTANCE,'/1X,
     +      'WITH EQUAL APRIORI PROBABILITIES')
      GOTO 20
   15 WRITE(21,219)
  219 FORMAT(///1X,'VALUES OF DIV. FN. FOR DIFFERENT CLASS-PAIRS'/
     +               1X,'WITH ASSN. OF EQUAL APRIORI PROBS.')
      WRITE(22,229)
  229 FORMAT(///1X,'FEATURE ORDERING BY DIVERGENCE FUNCTION,'/1X,
     +      'WITH EQUAL APRIORI PROBABILITIES')
   20 WRITE(6,601)
  601 FORMAT(1X,'ENTER NO. OF FEATURES: N')
      READ(5,*)N
      WRITE(6,602)
  602 FORMAT(1X,'ENTER NO. OF CLASSES: M')
      READ(5,*)M
      WRITE(6,603)
  603 FORMAT(1X,'ENTER NO. OF SAMPLING POINTS: NPNTS')
      READ(5,*)NPNTS
      WRITE(6,*)' DO YOU NEED TO IGNORE SOME CLASS-PAIRS?'
      READ(5,*)IGNORE
      PAIRS=M*(M-1)/2
      IF(IGNORE.EQ.'YES')THEN
      PAIRS=PAIRS-3.0
      ENDIF
C
C  FOR EACH OF THE N FEATURES DETERMINE THE VALUE OF THE
C  SPECIFIED CRITERION FUNCTION
C
C
      DO 1 K=1,N
```

```
C
C  READ THE (ESTIMATES OF) M CLASS-CONDITIONAL PROBABILITIES
C  FOR THE FEATURE UNDER CONSIDERATION FROM INF
C
      DO 2 I=1,M
      READ(10,101)(P(I,U),U=1,NPNTS)
  101 FORMAT(9X,10F8.4)
    2 CONTINUE
C
C
C  COMPUTE THE VALUES OF THE SPECIFIED DISTANCE CRITERION
C  BETWEEN VARIOUS CLASS-PAIRS
C
      DO 3 I=1,M-1
      DO 4 J=I+1,M
      GOTO(31,32,33,34,35),ICRIT
   31 D(I,J)=PERRS(P,M,MAXM,NPNTS,MAXPNTS,I,J)
      GOTO 40
   32 D(I,J)=KOLMS(P,M,MAXM,NPNTS,MAXPNTS,I,J)
      GOTO 40
   33 D(I,J)=BHATS(P,M,MAXM,NPNTS,MAXPNTS,I,J)
      GOTO 40
   34 D(I,J)=MATUS(P,M,MAXM,NPNTS,MAXPNTS,I,J)
      GOTO 40
   35 D(I,J)=DIVER(P,M,MAXM,NPNTS,MAXPNTS,I,J)
   40 CONTINUE
    4 CONTINUE
    3 CONTINUE
C
C  WRITE THE VALUES OF THE CRITERION FUNCTION UNDER CONSIDERATION
C  ON THE OUTPUT FILE OUTF1
C
      WRITE(21,211)K
  211 FORMAT(1X,'FEATURE NO. =',I5)
      DO 5 I=1,M-1
      WRITE(21,212)(D(I,J),J=I+1,M)
  212 FORMAT(1X,6E13.6)
    5 CONTINUE
C
C  IGNORE THE VALUES (OF THE CRITERION FUNCTION) FOR CLASS-PAIRS
C  REPRESENTING THE SAME NUMERAL
C
      IF(IGNORE.EQ.'YES')THEN
      D(2,3)=0
      D(6,7)=0
      D(10,11)=0
      ENDIF
C
C  (FOR THE FEATURE UNDER CONSIDERATION) COMPUTE THE AVERAGE VALUE
C  OF THE CRITERION FUNCTION OVER ALL THE CLASS-PAIRS (EXCEPT THOSE
C  REPRESENTING THE SAME NUMERAL)
C
      AVD(K)=0
      DO 6 I=1,M-1
      DO 7 J=I+1,M
```

```
      AVD(K)=AVD(K)+D(I,J)
    7 CONTINUE
    6 CONTINUE
      AVD(K)=AVD(K)/PAIRS
      IF(ICRIT.EQ.1.OR.ICRIT.EQ.3)THEN
      TEMP=-AVD(K)
      AVD(K)=TEMP
      ENDIF
C
    1 CONTINUE
C
C
C  ARRANGE THE FEATURES IN DECREASING ORDER OF EFFECTIVENESS
C BY ARRANGING THE VALUES OBTAINED ABOVE (USING THE 'NAG'
C  ROUTINE M01AKF)
C
      NW=N
      IFAIL=1
      CALL M01AKF(AVD,AVDW,ILOC,ILOCW,N,NW,IFAIL)
      IF(ICRIT.EQ.1.OR.ICRIT.EQ.3)THEN
      DO 9 K=1,N
      AVD(K)=-AVD(K)
    9 CONTINUE
      ENDIF
C
C  WRITE THE FEATURE NUMBERS IN THE ORDER OBTAINED ABOVE TOGETHER
C  WITH THEIR CRITERION-VALUES)
C
      WRITE(22,221)(AVD(K),ILOC(K),K=1,N)
  221 FORMAT(4(1X,E13.6,2X,I4))
C
      CLOSE(21)
      CLOSE(22)
      STOP
      END
C
  FUNCTION PERRS(P,M,MAXM,NPNTS,MAXPNTS,IROW1,IROW2)
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE BAYESIAN PROBABILITY
C   OF ERROR BETWEEN TWO ARRAYS OF PROBABILITIES
C
      REAL P(MAXM,MAXPNTS)
      PERRS=0
      DO 1 IU=1,NPNTS
      PERRS=PERRS+MIN(P(IROW1,IU),P(IROW2,IU))
    1 CONTINUE
      PERRS=0.50*PERRS
      END
C
  FUNCTION KOLMS(P,M,MAXM,NPNTS,MAXPNTS,IROW1,IROW2)
      REAL KOLMS
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE KOLMOGOROV
C  VARIATIONAL DISTANCE BETWEEN TWO ARRAYS OF PROBABILITIES
C
```

```
      REAL P(MAXM,MAXPNTS)
      KOLMS=0
      DO 1 IU=1,NPNTS
      KOLMS=KOLMS+ABS(P(IROW1,IU)-P(IROW2,IU))
    1 CONTINUE
      KOLMS=0.25*KOLMS
      END
C
  FUNCTION BHATS(P,M,MAXM,NPNTS,MAXPNTS,IROW1,IROW2)
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE BHATTACHARYYA
C  COEFFICIENT BETWEEN TWO ARRAYS OF PROBABILITIES
C
      REAL P(MAXM,MAXPNTS)
      BHATS=0
      DO 1 IU=1,NPNTS
      BHATS=BHATS+SQRT(P(IROW1,IU)*P(IROW2,IU))
    1 CONTINUE
      END
C
  FUNCTION MATUS(P,M,MAXM,NPNTS,MAXPNTS,IROW1,IROW2)
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE MATUSITA
C  DISTANCE BETWEEN TWO ARRAYS OF PROBABILITIES
C
      REAL P(MAXM,MAXPNTS),MATUS
      BHATS=0
      DO 1 IU=1,NPNTS
      BHATS=BHATS+SQRT(P(IROW1,IU)*P(IROW2,IU))
    1 CONTINUE
      MATUS=SQRT(2.0*(1.0-BHATS))
      END
C
  FUNCTION DIVER(P,M,MAXM,NPNTS,MAXPNTS,IROW1,IROW2)
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE DIVERGENCE FUNCTION
C  BETWEEN TWO ARRAYS OF PROBABILITIES
C
      REAL P(MAXM,MAXPNTS)
      DATA ZEROLOG/-6.192/
      DIVER=0.0
      DO 1 IU=1,NPNTS
      P1=P(IROW1,IU)
      P2=P(IROW2,IU)
      DIFF=P1-P2
      IF(ABS(DIFF).GE.0.0001)THEN
      IF(P1.GE.0.01)THEN
      P1LOG=LOG(P1)
      ELSE
      P1LOG=ZEROLOG
      ENDIF
      IF(P2.GE.0.01)THEN
      P2LOG=LOG(P2)
      ELSE
      P2LOG=ZEROLOG
```

```
      ENDIF
      DIVER=DIVER+(P1-P2)*(P1LOG-P2LOG)
      ENDIF
    1 CONTINUE
      END
```

APPENDIX B8

```
C
C  MCLASS:
C  FOR EACH OF A NO. OF FEATURES THIS PROGRAM CALCULATES THE VALUES
C  OF A SPECIFIED PROBABILISTIC DISTANCE CRITERION, AND THEN
C  ARRANGES THE FEATURES IN ASCENDING/DESCENDING ORDER OF THESE
C  VALUES
C
C   NOTE: THIS PROGRAM WORKS FOR BOTH EQUAL AND UNEQUAL APRIORI
C         PROBABILITIES
C
      PROGRAM MCLASS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+                    INF,OUTF,TAPE10=INF,TAPE20=OUTF)
C
C  INPUT FILE:
C                INF: CLASS-CONDITIONAL PROBABILITIES OF DIFFERENT
C                     FEATURES
C
C  OUTPUT FILE:
C               OUTF: FEATURES ARRANGED IN DECREASING ORDER OF THEIR
C                     EFFECTIVENESS AND THE VALUES OF THE FEATURE
C                     EVALUATION CRITERION USED
C
      DIMENSION P(13,10),AP(13),D(78),DW(78),ILOC(78),ILOCW(78)
      INTEGER U
      REAL MATUMS
      CHARACTER PRIOR*3
      CHARACTER HDR1*27,HDR2*13,HDR3*13,HDR4*22
      CHARACTER *32 HDR51,HDR52,HDR53,HDR54
      PARAMETER(HDR1='FEATURE ORDERING BY M-CLASS')
      PARAMETER(HDR2='WITH EQUAL    ')
      PARAMETER(HDR3=' WITH UNEQUAL')
      PARAMETER(HDR4=' APRIORI PROBABILITIES')
      PARAMETER(HDR51=' BAYESIAN PROBABILITY OF ERROR ')
      PARAMETER(HDR52=' AFFINITY MEASURE OF MATUSITA  ')
      PARAMETER(HDR53=' CONDITIONAL ENTROPY OF SHANNON')
      PARAMETER(HDR54=' BAYESIAN DISTANCE OF DEVIJVER ')
C
      REWIND 10
      REWIND 20
C
      WRITE(6,605)
  605 FORMAT(1X,'REFER TO THE FOLLOWING CODE-LIST TO SELECT THE'/
+      1X,'DESIRED M-CLASS DISTANCE CRITERION:')
      WRITE(6,606)
  606 FORMAT(/12X,'BAYESIAN PROBABILITY OF ERROR :  1'/1X,
+      11X,'MATUSITA''S MEASURE OF AFFINITY:  2  '/
+              12X,'SHANNON''S CONDITIONAL ENTROPY:    3'/
+              12X,'BAYESIAN DISTANCE OF DEVIJVER:    4')
C
C  SPECIFY THE FEATURE EVALUATION CRITERION TO BE USED
```

```
C
      WRITE(6,607)
  607 FORMAT(//1X,'ENTER THE CRITERION ID. CODE: ICRIT')
      READ(5,*)ICRIT
      WRITE(6,601)
  601 FORMAT(1X,'ENTER NO. OF FEATURES: N ')
      READ(5,*)N
      WRITE(6,602)
  602 FORMAT(1X,'ENTER NO. OF CLASSES: M ')
      READ(5,*)M
      WRITE(6,603)
  603 FORMAT(1X,'ENTER NO. OF SAMPLING POINTS: NPNTS ')
      READ(5,*)NPNTS
      WRITE(6,608)
  608 FORMAT(1X,'ARE THE APRIORI PROBS. UNEQUAL ?')
      READ(5,501)PRIOR
 501 FORMAT(A3)
      IF(PRIOR.EQ.'YES')THEN
      WRITE(6,604)
  604 FORMAT(1X,'ENTER APRIORI PROBABILITIES: AP(I)''S')
      READ(5,*)(AP(I),I=1,M)
      GOTO(11,12,13,14),ICRIT
   11 WRITE(20,212)HDR1,HDR51,HDR3,HDR4
      GOTO 20
   12 WRITE(20,212)HDR1,HDR52,HDR3,HDR4
  212 FORMAT(///2X,A27,A32/2X,A13,A22)
      GOTO 20
   13 WRITE(20,212)HDR1,HDR53,HDR3,HDR4
      GOTO 20
   14 WRITE(20,212)HDR1,HDR54,HDR3,HDR4
C
      ELSE
      REALM=M
      DO 10 I=1,M
      AP(I)=1.0/REALM
   10 CONTINUE
      GOTO(21,22,23,24),ICRIT
   21 WRITE(20,212)HDR1,HDR51,HDR2,HDR4
      GOTO 20
   22 WRITE(20,212)HDR1,HDR52,HDR2,HDR4
      GOTO 20
   23 WRITE(20,212)HDR1,HDR53,HDR2,HDR4
      GOTO 20
   24 WRITE(20,212)HDR1,HDR54,HDR2,HDR4
C
      ENDIF
C
C  FOR EACH OF THE N FEATURES DETERMINE THE VALUE OF THE SPECIFIED
C  CRITERION FUNCTION
C
   20 DO 1 K=1,N
C
C  READ THE (ESTIMATES OF) M CLASS-CONDITIONAL PROBABILITIES FOR
C  THE FEATURE UNDER CONSIDERATION FROM INF
C
```

```
      DO 2 I=1,M
      READ(10,101)(P(I,U),U=1,NPNTS)
  101 FORMAT(9X,10F8.4)
    2 CONTINUE
C
C  COMPUTE THE VALUE OF THE CRITERION FUNCTION UNDER CONSIDERATION
C
      GOTO(31,32,33,34),ICRIT
   31 D(K)=PERRMS(M,NPNTS,AP,P)
      GOTO 30
   32 D(K)=MATUMS(M,NPNTS,AP,P)
      GOTO 30
   33 D(K)=EQUIMS(M,NPNTS,AP,P)
      GOTO 30
   34 D(K)=DEVIMS(M,NPNTS,AP,P)
      GOTO 1
   30 D(K)=-D(K)
C
    1 CONTINUE
C
C  ARRANGE THE FEATURES IN DECREASING ORDER OF EFFECTIVENESS
C  BY ARRANGING THE CRITERION-VALUES OBTAINED ABOVE
C
      NW=N
      IFAIL=1
      CALL M01AKF(D,DW,ILOC,ILOCW,N,NW,IFAIL)
      IF(ICRIT.NE.4)THEN
      DO 9 K=1,N
      D(K)=-D(K)
    9 CONTINUE
      ENDIF
C
C  WRITE THE FEATURE NUMBERS IN THE ORDER OBTAINED ABOVE, TOGETHER
C  WITH THEIR CRITERION-VALUES, ON OUTF
C
      WRITE(20,201)(D(K),ILOC(K),K=1,N)
  201 FORMAT(4(1X,E13.6,2X,I4))
C
      CLOSE(20)
      STOP
      END
C
  FUNCTION PERRMS(M,NPNTS,AP,P)
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE BAYESISN ERROR
C  PROBABILITY IN AN M-CLASS SITUATION
C
      DIMENSION AP(M),P(M,NPNTS)
      INTEGER U
      PERRMS=1.0
      DO 1 U=1,NPNTS
      PMAX=AP(1)*P(1,U)
      DO 2 I=2,M
      PROD=AP(I)*P(I,U)
      IF(PMAX.LT.PROD)THEN
```

```
      PMAX=PROD
      ENDIF
    2 CONTINUE
      PERRMS=PERRMS-PMAX
    1 CONTINUE
      END
C
  FUNCTION MATUMS(M,NPNTS,AP,P)
C
C  THIS FUNCCTION SUBPROGRAM CALCULATES THE M-CLASS
C  AFFINITY OF MATUSITA
C
      REAL MATUMS, AP(M),P(M,NPNTS)
      INTEGER U
C
      POWER=1.0/REAL(M)
      EPS=1.0E-260
      TERM1=1.0
      DO 1 I=1,M
      TERM1=TERM1*AP(I)
    1 CONTINUE
      IF(TERM1.GT.EPS)THEN
      TERM1=TERM1**POWER
      ELSE
      TERM1=0
      ENDIF
      TERM2=0
      DO 3 U=1,NPNTS
      PROD=1
      DO 4 I=1,M
      PROD=PROD*P(I,U)
    4 CONTINUE
      IF(PROD.GT.EPS)THEN
      PROD=PROD**POWER
      TERM2=TERM2+PROD
      ENDIF
    3 CONTINUE
      MATUMS=TERM1*TERM2
      END
C
  FUNCTION EQUIMS(M,NPNTS,AP,P)
C
C  THIS FUNCTION SUBPROGRAM CALCULATES THE M-CLASS CONDITIONAL
C   ENTROPY ( EQUIVOCATION ) OF SHANNON
C
      REAL AP(M),P(M,NPNTS)
      INTEGER U
      EPS=1.0E-260
      EQUIMS=0
      DO 1 U=1,NPNTS
      TERM1=0
      DO 2 I=1,M
      TERM1=TERM1+AP(I)*P(I,U)
    2 CONTINUE
      IF(TERM1.GT.EPS)THEN
```

```
        TERM1=TERM1*LOG(TERM1)
        ELSE
        TERM1=0
        ENDIF
        TERM2=0
        DO 3 I=1,M
        PROD=AP(I)*P(I,U)
        IF(PROD.GT.EPS)THEN
        TERM2=TERM2+PROD*LOG(PROD)
        ENDIF
      3 CONTINUE
        EQUIMS=EQUIMS+TERM1-TERM2
      1 CONTINUE
        END
C
    FUNCTION DEVIMS(M,NPNTS,AP,P)
C
C    THIS FUNCTION SUBPROGRAM CALCULATES M-CLASS BAYESIAN
C    DISTANCE OF DEVIJVER
C
        REAL AP(M),P(M,NPNTS)
        INTEGER U
        EPS=1.0E-260
        DEVIMS=0
        DO 1 U=1,NPNTS
        XNUM=0
        XDEN=0
        DO 2 I=1,M
        PROD=AP(I)*P(I,U)
        XDEN=XDEN+PROD
        PRODSQ=PROD*PROD
        XNUM=XNUM+PRODSQ
      2 CONTINUE
        IF(XDEN.GT.EPS)THEN
        DEVIMS=DEVIMS+XNUM/XDEN
        ENDIF
      1 CONTINUE
        END
```

APPENDIX B9

```
C
C  MAHAL1:
C  FOR EACH OF A NO. OF FEATURES THIS PROGRAM CALCULATES THE VALUES
C  TAKEN BY MAHALANOBIS DISTANCE BETWEEN DIFFERENT CLASS-PAIRS(WITH
C  ASSUMPTION OF INDEPENDENT FEATURES), ITS AVERAGE OVER ALL
C  CLASS-PAIRS, AND THEN ARRANGES THE FEATURES IN DESCENDING ORDER
C  OF THE AVERAGE VALUES
C
      PROGRAM MAHAL1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+             INF,OUTF1,OUTF2,TAPE10=INF,TAPE21=OUTF1,TAPE22=OUTF2,
+             OUTF3,TAPE23=OUTF3)
C
C  NOTE: APRIORI PROBABILITIES ARE TAKEN TO BE EQUAL
C
C  INPUT FILE:
C                  INF:  ARITHMETIC MEANS AND STANDARD DEVIATIONS OF
C                        DIFFERENT FEATURES IN DIFFERENT CLASSES
C
C  OUTPUT FILES:
C                 OUTF1: VALUES OF THE SAMPLE MAHALANOBIS DISTANCE
C                        BETWEEN VARIOUS CLASS-PAIRS FOR DIFFERENT
C                        FEATURES
C                 OUTF2: FEATURES ARRANGED IN DECREASING ORDER OF THEIR
C                        AVERAGE CRITERION VALUES (TOGETHER WITH THESE
C                        CRITERION-VALUES)
C                 OUTF3: FEATURES WITH VERY SMALL VARIANCE-VALUES
C
      DIMENSION NOBS(13),AM(13,78),SD(13,78),AVD(78),AVDW(78)
      DIMENSION ILOC(78),ILOCW(78)
      DIMENSION XNOBS(13)
C
      CHARACTER*3 IGNORE
C  STATEMENT FUNCTION DEFINING THE CRITERION
      FUNC(DD)=1.0-EXP(-DD/8.0)
C
      REWIND 10
      REWIND 21
      REWIND 22
      REWIND 23
C
      WRITE(22,222)
  222 FORMAT(///1X,'FEATURE ORDERING BY MAHALANOBIS DISTANCE'/
+               1X,'WITH ASSUMPTION OF INDEPENDENT FEATURES')
      WRITE(21,212)
  212 FORMAT(///1X,'FEATURE',2X,'CL1',3X,'CL2',5X,'VARIANCE',1X,
+             'MAHAL. DIST.')
      WRITE(6,600)
  600 FORMAT(1X,'ENTER NO. OF FEATURES ?')
      READ(5,*)N
      WRITE(6,601)
```

```
  601 FORMAT(1X,'ENTER NO. OF CLASSES ?')
      READ(5,*)M
      WRITE(6,603)
  603 FORMAT(1X,'IS THERE ANY REDUCTION IN NO. OF CLASS-PAIRS?')
      READ(5,500)IGNORE
  500 FORMAT(A3)
      WRITE(6,602)
  602 FORMAT(1X,'ENTER NO. OF OBS. IN DIFFERENT CLASSES ?')
      READ(5,*)(NOBS(I),I=1,M)
C
C  READ THE ARITHMETIC MEANS AND STANDARD DEVIATIONS FROM INF
C
      DO 1 I=1,M
      DO 2 K=1,N
      READ(10,100)AM(I,K),SD(I,K)
  100 FORMAT(11X,F10.4,1X,E14.7)
    2 CONTINUE
    1 CONTINUE
C
      DO 8 I=1,M
      XNOBS(I)=NOBS(I)
    8 CONTINUE
C
      EPS=1.0E-260
C
C  CALCULATE THE EFFECTIVE NUMBER OF CLASS-PAIRS
C
      PAIRS=M*(M-1)/2
      IF(IGNORE.EQ.'YES')THEN
      PAIRS=PAIRS-3.0
      ENDIF
C
C  FOR EACH OF THE N FEATURES COMPUTE THE VALUES OF THE SPECIFIED
C  CRITERION AND THEN AVERAGE THEM OVER THE EFFECTIVE NO. OF CLASS-PAIRS
C
      DO 3 K=1,N
      AVD(K)=0
      DO 4 I=1,M-1
      DO 5 J=I+1,M
      IF(IGNORE.EQ.'YES')THEN
C
C  IGNORE CLASS-PAIR (I,J) IF THEY REPRESENT THE SAME CLASS
C
      IF((I.EQ.2.AND.J.EQ.3).OR.(I.EQ.6.AND.J.EQ.7).OR.
+         (I.EQ.10.AND.J.EQ.11))THEN
      WRITE(23,230)K,I,J
  230 FORMAT(1X,'FEATURE',I4,' CLASS-PAIR:',2I3,' :COMP. IGNORED')
      GOTO 5
      ELSE
      SSI=(XNOBS(I)-1.0)*SD(I,K)**2
      SSJ=(XNOBS(J)-1.0)*SD(J,K)**2
      VAR=(SSI+SSJ)/(XNOBS(I)+XNOBS(J)-2.0)
      DIFF2=(AM(I,K)-AM(J,K))**2
      IF(DIFF2.GE.EPS)THEN
      IF(VAR.LT.EPS)THEN
```

```
      AVD(K)=AVD(K)+1.0
C
C  WRITE OUTF3
C
      WRITE(23,231)K,I,J,DIFF2,VAR
  231 FORMAT(1X,'FEATURE',I4,' CLASS-PAIR:',2I3,' NUMERATOR=',
+      E13.6,' DENOMINATOR=',E13.6/
+      1X,'DENOMINATOR TOO SMALL, VALUE OF CRITERION= 1 ')
      GOTO 5
      ELSE
      D2=DIFF2/VAR
C
C  WRITE OUTF1
C
      WRITE(21,211)K,I,J,VAR,D2
  211 FORMAT(1X,3I6,2E13.6)
      AVD(K)=AVD(K)+FUNC(D2)
      ENDIF
      ELSE
      WRITE(23,232)K,I,J,DIFF2
  232 FORMAT(1X,'FEATURE',I4,' CLASS-PAIR:',2I3,' NUMERATOR=',
+      E13.6/1X,'NUMERATOR TOO SMALL, VALUE OF CRITERION= 0 ')
      ENDIF
      ENDIF
      ENDIF
    5 CONTINUE
    4 CONTINUE
      AVD(K)=AVD(K)/PAIRS
    3 CONTINUE
C
C  ARRANGE THE FEATURES IN ORDER USING THE 'NAG' ROUTINE M01AKF
C
      NW=N
      IFAIL=1
      CALL M01AKF(AVD,AVDW,ILOC,ILOCW,N,NW,IFAIL)
C
C  WRITE THE FEATURE NUMBERS IN THE ORDER OBTAINED ABOVE, TOGETHER
C  WITH THEIR AVERAGE CRITERION-VALUES, ON OUTF2
C
      WRITE(22,221)(AVD(K),ILOC(K),K=1,N)
  221 FORMAT(4(1X,E13.6,2X,I4))
      CLOSE (21)
      CLOSE (22)
      CLOSE (23)
      STOP
      END
```

APPENDIX B10

```
C  MAHAL2:
C  THIS PROGRAM PERFORMS STEPWISE FEATURE SUBSET SELECTION IN A
C  MULTICLASS ENVIRONMENT USING TWO-CLASS MAHALANOBIS DISTANCE
C  CRITERION : D2/ (4 + D2)
C
C  NOTES: G-INVERSE ROUTINE USED, ALL CLASS-PAIRS CONSIDERED,
C         TO CHANGE THE CRITERION WE NEED CHANGE ONLY THE STATEMENT
C         FUNCTION
C
      PROGRAM MAHAL2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT, INF1,INF2,
+             INF3,OUTF1,OUTF2,OUTF3,TAPE11=INF1,TAPE12=INF2,TAPE13=INF3,
+             TAPE21=OUTF1,TAPE22=OUTF2,TAPE23=OUTF3)
C
C
C  INPUT FILES:
C                 INF1:  NO. OF OBSERVATIONS IN DIFFERRENT CLASSES
C                 INF2:  INITIAL ORDERING OF FEATURES
C                 INF3:  MEANS AND COVARIANCES OF FEATURES IN DIFFERENT
C                        CLASSES
C
C  OUTPUT FILES:
C                 OUTF1: FEATURE SUBSETS SELECTED IN STEPS 1,2,...,N
C                 OUTF2: ABOVE FEATURE SUBSETS TOGETHER WITH THE
C                        CORRESPONDING CRITERION-VALUES
C                 OUTF3: BOTH SELECTED AND UNSELECTED FEATURE SUBSETS
C                        WITH THEIR CRITERION-VALUES
C
      DIMENSION NOBS(2),AM(2,25),COV(2,25,25),CRIT(25),
+      KSEL(25),NSEL(25),XIBAR(25),XJBAR(25),SI(25,25),SJ(25,25),
+      DIFF(25),SIJ(25,25),KOLD(25),AIJMX(25),D(25),U(25,25),
+      DU(25),INC(25),XNOBS(2)
C
      CHARACTER*3 IGNORE
C
      DATA IA/25/ , IU/25/
      DATA NMAX / 25 /
C
C  STATEMENT FUNCTION DEEFINING THE CRITERION
C
      FUNC(DD)=DD/(4.0+DD)
C
      REWIND 11
      REWIND 12
      REWIND 13
      REWIND 21
      REWIND 22
      REWIND 23
      REWIND 24
C
C  FOLLOWING PARAMETERS ARE REQUIRED BY THE G-INVERSE ROUTINE
```

```
      EPS=X02AAF(4321)
      IFAIL=0
C
      WRITE(24,*)'VALUES OF: KSTEP,KSET,I,J,DIJ2,T,CONDITION NO.'
C
      WRITE(6,600)
  600 FORMAT(1X,'ENTER NO. OF CLASSES')
      READ(5,*)M
      WRITE(6,601)
  601 FORMAT(1X,'ENTER NO. OF FEATURES')
      READ(5,*)N
      WRITE(6,602)
  602 FORMAT(1X,'IS THERE ANY REDUCTION IN NO. OF CLASS-PAIRS?')
      READ(5,500)IGNORE
  500 FORMAT(A3)
C
C  READ NO. OF OBS. IN DIFFERENT CLASSES FROM INF1
C
      READ(11,*)(NOBS(I),I=1,M)
C
      DO 22 I = 1,M
      XNOBS(I) = NOBS(I)
   22 CONTINUE
C
C
C  CALCULATE THE EFFECTIVE NO. OF CLASS-PAIRS
C
      PAIRS=M*(M-1)/2
      IF(IGNORE.EQ.'YES')THEN
      PAIRS = PAIRS-3.0
      ENDIF
C
C  READ THE MEAN VECTORS AND THE COVARIANCE MATRICES FROM INF3
C
      DO 1 I=1,M
      DO 2 K1=1,N
      READ(13,130)AM(I,K1),(COV(I,K1,K2),K2=K1,N)
  130 FORMAT(11X,F10.4/(1X,5E14.7))
    2 CONTINUE
    1 CONTINUE
C  FILL IN THE LOWER TRIANGLES OF THE COVARIANCE MATRICES
      DO 40 I=1,M
      DO 41 K1=2,N
      DO 42 K2=1,K1-1
      COV(I,K1,K2) = COV(I,K2,K1)
   42 CONTINUE
   41 CONTINUE
   40 CONTINUE
C
C  READ THE CORRESPONDENCE BETN. NEW AND OLD FEATURES FROM INF2
C
      READ(12,129)
  129 FORMAT()
      READ(12,120)(KOLD(K),K=1,N)
  120 FORMAT(6(6X,I4,3X))
```

```
C
C  INITIALIZE THE TWO INTEGER ARRAYS KSEL ( LIST OF FEATURES SELECTED
C  IN THE PREVIOUS STEP ) AND NSEL ( LIST OF FEATURES FROM WHICH THE
C  NEXT FEATURE TO BE SELECTED )
C
      DO 3 K = 1,N
      KSEL(K) = 0
      NSEL(K) = K
    3 CONTINUE
C
C  N FEATURE EVALUATION STEPS START HERE
C
      DO 4 KSTEP = 1, N
C
      IF ( KSTEP.GT.1 ) THEN
C
C  LOCATE THE FEATURES NOT YET SELECTED : ARRAY NSEL
C
      DO 5 KK = 1, N-KSTEP+1
      NSEL(KK) = 0
    5 CONTINUE
      KK=1
      DO 6 K = 1,N
      DO 7 L = 1, KSTEP-1
      IF ( K.EQ.KSEL(L)) GOTO 6
    7 CONTINUE
      DO 8 L = 1, KK
      IF ( K.EQ.NSEL(L)) GOTO 6
    8 CONTINUE
      NSEL(KK) = K
      KK = KK + 1
    6 CONTINUE
      ENDIF
C
C  WRITE STEP NO. IN OUTPUT FILE OUTF3
C
      WRITE(23,231)KSTEP
  231 FORMAT(1X,'SELECTION STEP : ',I4)
C
C  OBTAIN THE CRITERION VALUES FOR N - KSTEP + 1 FEATURE SUBSETS
C  EACH OF WHICH INCLUDES THE PREVIOUSLY SELECTED KSTEP -1 FEATURES
C
      DO 9 KSET = 1, N-KSTEP+1
      CRIT(KSET) = 0.0
C
      DO 10 I = 1,M-1
C
      DO 11 J = I+1, M
C
      IF(IGNORE.EQ.'YES')THEN
C
C  OMIT CLASS-PAIR ( I,J ) IF THEY REPRESENT THE SAME CLASS
C
      IF((I.EQ.2.AND.J.EQ.3).OR.(I.EQ.6.AND.J.EQ.7).OR.
+         (I.EQ.10.AND.J.EQ.11)) GOTO 11
```

```
C
      ENDIF
C
      IF ( KSTEP.GT.1 ) THEN
C
C  OBTAIN MEAN VECTORS AND COVARIANCE MATRICES OF PREVIOUSLY SELECTED
C  FEATURES FOR I-TH AND J-TH CLASSES
C
      DO 12 K1 = 1, KSTEP-1
      XIBAR(K1) = AM(I,KSEL(K1))
      XJBAR(K1) = AM(J,KSEL(K1))
C
      DO 13 K2 = K1, KSTEP-1
      SI(K1,K2) = COV(I,KSEL(K1),KSEL(K2))
      SJ(K1,K2) = COV(J,KSEL(K1),KSEL(K2))
   13 CONTINUE
C
C  OBTAIN COVARIANCES OF PREVIOUSLY SELECTED FEATURES WITH THE
C  NEW FEATURE
C
      SI(K1,KSTEP) = COV(I,KSEL(K1),NSEL(KSET))
      SJ(K1,KSTEP) = COV(J,KSEL(K1),NSEL(KSET))
C
   12 CONTINUE
C
      ENDIF
C
C  OBTAIN MEANS AND VARIANCES ( IN TWO CLASSES ) OF THE NEW FEATURE
C
      XIBAR(KSTEP) = AM(I,NSEL(KSET))
      XJBAR(KSTEP) = AM(J,NSEL(KSET))
      SI(KSTEP,KSTEP) = COV(I,NSEL(KSET),NSEL(KSET))
      SJ(KSTEP,KSTEP) = COV(J,NSEL(KSET),NSEL(KSET))
C
C  COMPUTE THE DIFFERENCE OF THE TWO MEAN VECTORS AND THE AVERAGE OF
C  THE TWO COVARIANCE MATRICES
C
      DO 14 K1 = 1, KSTEP
      DIFF(K1) = XIBAR(K1) - XJBAR(K1)
      DO 15 K2 = K1, KSTEP
      SIJ(K1,K2) =(SI(K1,K2) * (XNOBS(I)-1.0) + SJ(K1,K2) *
+      (XNOBS(J)-1.0)) / (XNOBS(I)+XNOBS(J)-2.0)
   15 CONTINUE
   14 CONTINUE
C
C  OBTAIN THE COVARIANCE VALUES IN THE LOWER TRIANGLE OF THE
C  COVARIANCE MATRIX BY TRANSPOSING THE UPPER TRIANGLE
C
      IF ( KSTEP.GT.1 ) THEN
      DO 16 K1 = 2, KSTEP
      DO 17 K2 = 1,K1-1
      SIJ(K1,K2) = SIJ(K2,K1)
   17 CONTINUE
   16 CONTINUE
C
```

```
      ENDIF
C
C  CHECK IF THE VARIANCE TERMS IN THE COVARIANCE MATRIX UNDER
C  CONSIDERATION ARE ALL EQUAL TO ZERO
C
      DO 18 K = 1, KSTEP
      IF ( SIJ(K,K).GT.EPS ) GOTO 19
   18 CONTINUE
C
C  IF ALL ARE ZERO THEN GO TO NEXT J ( CLASS J)
C
      GO TO 11
C
C  OTHERWISE CALCULATE THE G-INVERSE OF THE COVARIANCE MATRIX SIJ
C  USING THE NAG ROUTINE
C
   19 CONTINUE
C
C  DETERMINE THE NORM OF THE MATRIX SIJ (SNORM) AND THEN DECIDE T
C
      DO 31 K1=1,KSTEP
      D(K1)=0
      DO 32 K2=1,KSTEP
      D(K1)=D(K1)+ABS(SIJ(K1,K2))
   32 CONTINUE
   31 CONTINUE
      LOCNORM=LOCMAX(KSTEP,NMAX,D)
      VALNORM=D(LOCNORM)
      CXIX=REAL(KSTEP)*VALNORM
      T=CXIX*EPS
    CALL F01BLF(KSTEP,KSTEP,T,SIJ,IA,AIJMX,IRANK,INC,D,U,IU,DU,IFAIL)
C
C  COMPUTE MAHALANOBIS DISTANCE ( DIJ2 ) BETWEEN I-TH AND J-TH CLASSES
C  USING FUNCTION SUBPROGRAM QUADR
C
      DIJ2 = QUADR ( KSTEP,NMAX,DIFF,SIJ )
C
C  ADD THE CONTRIBUTION OF DIJ2 TO THE CRITERION VALUE
C
      CRIT (KSET) = CRIT (KSET) + FUNC(DIJ2)
C
   11 CONTINUE
   10 CONTINUE
C
C  COMPUTE THE AVERAGE CRITERION VALUE
C
      CRIT (KSET) = CRIT (KSET) / PAIRS
C
C  CREATE FILE OUTF3 WHICH CONTAINS VERY DETAILED OUTPUT
C
      WRITE(23,232)
  232 FORMAT(1X,'FEATURE SET UNDER CONSIDERATION : ')
C
      IF ( KSTEP.GT.1 ) THEN
      WRITE(23,233)(KOLD(KSEL(K)),K=1,KSTEP-1),KOLD(NSEL(KSET))
```

```
  233 FORMAT(1X,13I6)
      ELSE
      WRITE(23,233)KOLD(NSEL(KSET))
      ENDIF
C
      WRITE(23,234) CRIT(KSET)
  234 FORMAT(1X,'CRITERION VALUE = ',E14.7)
C
C
    9 CONTINUE
C
C
C  SELECT THE FEATURE SET ( OF SIZE KSTEP ) WITH MAXIMUM CRITERION
C  VALUE USING FUNCTION SUBPROGRAM LOCMAX
C
      MAXK = LOCMAX ( N-KSTEP+1,NMAX,CRIT )
      KSEL (KSTEP) = NSEL (MAXK)
C
C  CREATE FILE OUTF2 WHICH CONTAINS SELECTED FEATURE SET AND THE
C  CORRESPONDING CRITERION VALUE
C
      WRITE(22,221) KSTEP
  221 FORMAT(5X,'SELECTED FEATURES IN STEP : ',I6)
      WRITE(22,222) ( KOLD(KSEL(K)),K=1,KSTEP )
  222 FORMAT ( 1X, 13I6 )
      WRITE(22,223) CRIT(MAXK)
  223 FORMAT(1X,'CRITERION VALUE = ', E14.7)
C
C  CREATE FILE OUTF1 CONTAINING LIST OF SELECTED FEATURES ( OUTF1 IS
C  IS A SUBSET OF OUTF2 )
C
      WRITE(21,211) KSTEP
  211 FORMAT(5X,'SELECTED FEATURES IN STEP : ',I6)
      WRITE(21,212) (KOLD(KSEL(K)),K=1,KSTEP)
  212 FORMAT( 1X,13I6 )
C
C
C
    4 CONTINUE
C
C
C
      CLOSE(21)
      CLOSE(22)
      CLOSE(23)
C
      STOP
      END
C
  FUNCTION QUADR (N,NMAX,X,A)
C
C  THIS FUNCTION SUBPROGRAM DETERMINES THE QUADRATIC FORM OF
C  A VECTOR AND A MATRIX
C
      DIMENSION X(NMAX),A(NMAX,NMAX)
```

```
C
      QUADR=0.0
      DO 1 I=1,N
      DO 2 J=1,N
      QUADR=QUADR+X(I)*X(J)*A(I,J)
    2 CONTINUE
    1 CONTINUE
      END
C
  FUNCTION LOCMAX (N,NMAX,X)
C
C  THIS FUNCTION SUBPROGRAM DETERMINES THE LOCATION OF THE
C  MAXIMUM ELEMENT OF AN ARRAY
C
      DIMENSION X(NMAX)
C
      LOCMAX=1
      XMAX=X(1)
      IF (N.GT.1) THEN
      DO 1 K = 2,N
      IF (XMAX.LT.X(K)) THEN
      XMAX=X(K)
      LOCMAX = K
      ENDIF
    1 CONTINUE
      ENDIF
      END
```

APPENDIX B11

```
C
C  RANKCOR:
C      THIS PROGRAM COMPUTES THE RANK CORRELATION BETWEEN TWO SETS
C      OF FEATURE ORDERINGS
C
      PROGRAM RANKCOR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,INF1,INF2,
+                     TAPE11=INF1,TAPE12=INF2,OUTF,TAPE20=OUTF)
C
C  INPUT FILES:
C                 INF1: FIRST SET OF FEATURE ORDERINGS
C                 INF2: SECOND SET OF FEATURE ORDERINGS
C
C  OUTPUT FILE:
C                 OUTF: VALUE OF THE SPECIFIED RANK CORRELATION
C                       COEFFICIENT (ALSO INCLUDED THE TWO SETS OF
C                       FEATURE ORDERINGS SUPPLIED AS INPUT)
C
      DIMENSION IR1(78),IR2(78)
      REAL KENDAL
      DATA NMAX/78/
C
      REWIND 11
      REWIND 12
      REWIND 20
C
      WRITE(6,600)
  600 FORMAT(1X,'REFER TO THE FOLLOWING LIST TO SELECT'/
+             1X,'THE CORRELATION CRITERION')
      WRITE(6,601)
  601 FORMAT(/12X,'KENDAL''S TAU-CRITERION    : 1'/
C
C  SPECIFY THE RANK CORRELATION COEFFICIENT TO BE USED
C
+              12X,'SPEARMAN''S RHO-CRITERION : 2')
      WRITE(6,602)
  602 FORMAT(//1X,'ENTER CRITERION IDENTIFICATION CODE: ICRIT')
      READ(5,*)ICRIT
C
      WRITE(6,603)
  603 FORMAT(1X,'ENTER THE NO. OF FEATURES: N')
      READ(5,*)N
C
C  READ THE TWO SETS OF FEATURE ORDERINGS
C
      READ(11,110)(IR1(I),I=1,N)
 110 FORMAT(//////(4(16X,I4)))
      READ(12,120)(IR2(I),I=1,N)
 120  FORMAT(//////(4(16X,I4)))
C
C  WRITE THE TWO SETS OF FEATURE ORDERINGS ON OUTF
```

```
C
      WRITE(20,200)
  200 FORMAT(1X,'SET 1 :')
      WRITE(20,201)(IR1(I),I=1,N)
  201 FORMAT(1X,6(13I6/))
      WRITE(20,202)
  202 FORMAT(1X,'SET 2 :')
      WRITE(20,201)(IR2(I),I=1,N)
C
C  COMPUTE THE CRITERION VALUE AND WRITE IT ON OUTF
C
      GOTO(1,2),ICRIT
    1 CORR=KENDAL(N,NMAX,IR1,IR2)
      WRITE(20,204)CORR
  204 FORMAT(//1X,'KENDAL''S CORR. COEFF. =',E14.6)
      WRITE(6,204)CORR
      CLOSE (20)
      STOP
    2 CORR=SPEAR(N,NMAX,IR1,IR2)
      WRITE(20,205)CORR
  205 FORMAT(//1X,'SPEARMAN''S CORR. COEFF. =',E14.6)
      WRITE(6,205)CORR
      CLOSE(20)
      STOP
      END
C
 FUNCTION KENDAL(N,NMAX,IR1,IR2)
C
C   THIS FUNCTION SUBPROGRAM COMPUTES KENDAL'S RANK CORRELATION
C  COEFFICIENT, USUALLY DENOTED BY GREEK LETTER TOU
C
      DIMENSION IR1(NMAX),IR2(NMAX)
      REAL KENDAL
C
      SUM=0
      DO 1 I=1,N-1
      DO 2 J=I+1,N
      IPROD=(IR1(I)-IR1(J))*(IR2(I)-IR2(J))
      IF(IPROD.GT.0)THEN
      SUM=SUM+1.0
      GOTO 2
      ELSE IF(IPROD.LT.0)THEN
      SUM=SUM-1.0
      ENDIF
    2 CONTINUE
    1 CONTINUE
      KENDAL=2.0*SUM/REAL(N*(N-1))
      END
C
  FUNCTION SPEAR(N,NMAX,IR1,IR2)
C
C   THIS FUNCTION SUBPROGRAM COMPUTES SPEARMAN'S RANK CORRELATION
C
      DIMENSION IR1(NMAX),IR2(NMAX)
C
```

```
      ISSQ=0
      DO 1 I=1,N
      ISSQ=ISSQ+(IR1(I)-IR2(I))**2
    1 CONTINUE
      SSQ=ISSQ
      DEN=N*N*N-N
      SPEAR=1.0-(6.0*SSQ)/DEN
      END
```

APPENDIX B12

```
C
C  RECOG1:
C          PROGRAM FOR RECOGNITION OF NUMERALS USING NONPARAMETRIC
C          BAYESIAN APPROACH WITH THE ASSUMPTION OF INDEPENDENCE OF
C          FEATURES
C
  PROGRAM RECOG1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+  INF1,INF2,INF3,INF4,INF5,TAPE11=INF1,TAPE12=INF2,TAPE13=INF3,
+  TAPE14=INF4,TAPE15=INF5,TAPE22)
C
C  INPUT FILES :
C                INF1: MIN. AND MAX. OF THE FEATURES
C                INF2: PDF'S OF CLASSES
C                INF3: FEATURE VALUES
C                INF4: FEATURES IN DESCENDING ORDER OF EFFECTIVENESS
C                INF5: APRIORI PROBABILITIES OF CLASSES
C
C  OUTPUT FILE :
C                TAPE22: CONFUSION MATRIX ( SAVED AS 'CONFMAT' )
C
C
      DIMENSION P(78,13,10), XMIN(78), XMAX(78), H(78), AP(13),
+               R(13),X(78), IFEAT(78), IFR(0:9,14), NFSIZE(40)
C
      CHARACTER PFCMD*80, MSG*40
      CHARACTER OUTFNM*7
C
C
      DATA M/13/, N/78/, NPNTS/10/, MMAX/13/
      DATA OUTFNM/'CONFMAT'/
C
     WRITE(6,*)'ENTER NO. OF FEATURE SETS: NFSETS'
     READ(5,*)NFSETS
     WRITL(6,*)'ENTER SIZES OF THESE FEATURE SETS'
     READ(5,*)(NFSIZE(KK),KK=1,NFSETS)
     WRITE(6,*)'ENTER NO. OF OBS. TO BE RECOGNIZED'
     READ(5,*)NOBS
C
      REWIND 11
      REWIND 12
      REWIND 15
C
C
C  READ THE APRIORI PROBABILITIES OF CLASSES FROM INF5
C
      READ(15,*)(AP(I),I=1,M)
C
C  READ THE ESTIMATED PROB. VALUES, IN DIFFERENT SAMPLING INTERVALS (J)
C  OF DIFFERENT FEATURES (K), FOR DIFFERENT CLASSES (I)
C
```

```
      DO 1 K=1,N
      DO 2 I=1,M
      READ(12,120)(P(K,I,J),J=1,NPNTS)
  120 FORMAT(9X,10F8.4)
    2 CONTINUE
    1 CONTINUE
C
C  READ THE MINIMUM AND MAXIMUM VALUES OF DIFFERENT FEATURES
C
      READ(11,110)(XMIN(K),XMAX(K),K=1,N)
  110 FORMAT(1X,8F8.4)
C
C  COMPUTE (SAMPLING INTERVAL)/2 FOR DIFFERENT FEATURES
C
      XNP=NPNTS
      DO 3 K = 1,N
      H(K) = (XMAX(K) - XMIN(K)) / (2.0*XNP)
    3 CONTINUE
C
      REWIND 22
C
C
       DO 17 KK = 1,NFSETS
C
C
       N1 = NFSIZE(KK)
C
       REWIND 13
       REWIND 14
C
C  READ THE FEATURE NO.S TO USE FOR RECOGNITION FROM INF4
C
       READ(14,141)(IFEAT(K1),K1=1,N1)
  141 FORMAT(/////(4(16X,I4)))
C
C  WRITE THE LIST OF FEATURES ON OUTF2
C
       WRITE(22,226)N1
  226 FORMAT(1X,'NO.OF FEATURES USED = ',I4//)
       WRITE(22,225)(IFEAT(K1),K1=1,N1)
  225 FORMAT(1X,'FEATURES USED: '/(15I5))
C
C  INITIALIZE THE CELLS OF THE CONFUSION MATRIX
C
      DO 10 I1=0,9
      DO 11 I2=1,M+1
      IFR(I1,I2)=0
   11 CONTINUE
   10 CONTINUE
C
C  RECOGNITION PROCESS STARTS HERE
C
      DO 14 L = 1,NOBS
C
       READ(13,130)NUMID,(X(K),K=1,N)
```

```
  130 FORMAT(2X,I4/(1X,8F8.4))
      ITRU = NUMID/1000
C
      DO 4 I = 1,M
      R(I) = AP(I)
    4 CONTINUE
C
      DO 5 K1 = 1,N1
C
      Y = X(IFEAT(K1))
      YMIN = XMIN(IFEAT(K1))
      HK1 = H(IFEAT(K1))
C
      IF(Y.LT.(YMIN+2.0*HK1))THEN
      INTV = 1
      GOTO 6
      ELSE IF (Y.GE.(YMIN+2.0*(XNP-1.0)*HK1))THEN
      INTV = NPNTS
      GOTO 6
      ELSE
      DO 7 J = 2,NPNTS-1
      XJ = J
      IF((Y.GE.(YMIN+2.0*(XJ-1.0)*HK1)).AND.(Y.LT.(YMIN+2.0*XJ*HK1)))THEN
      INTV = J
      GOTO 6
      ENDIF
    7 CONTINUE
      ENDIF
    6 CONTINUE
C
C  COMPUTE THE VALUES OF THE CLASSIFICATION CRITERION R(I)
C
      DO 8 I = 1,M
      R(I) = R(I)*P(IFEAT(K1),I,INTV)
    8 CONTINUE
C
    5 CONTINUE
C
C  OBTAIN THE CLASS WITH MAXIMUM R-VALUE
C
      IREC = LOCMAX(M,MMAX,R)
      IF(R(IREC).LT.1.0E-250)THEN
      IREC=14
      ENDIF
      IFR(ITRU,IREC) = IFR(ITRU,IREC) +1
C
   14 CONTINUE
C
C  COMPUTE THE RECOGNITION ACCURACY AND CREATE THE OUTPUT FILES
C
      CORCT=IFR(0,1)+IFR(1,2)+IFR(1,3)+IFR(2,4)+IFR(3,5)+IFR(4,6)+
     +  IFR(4,7)+IFR(5,8)+IFR(6,9)+IFR(7,10)+IFR(7,11)+IFR(8,12)+
     +  IFR(9,13)
      CRATE = CORCT/REAL(NOBS)*100.0
      WRITE(22,223)
```

```
  223 FORMAT(1X,'CONFUSION MATRIX :')
      WRITE(22,224)
  224 FORMAT(5X,'NUMERAL',2X,24('-'),'RECOGNIZED AS',27('-'))
      WRITE(22,222)
  222 FORMAT(14X,'0',4X,'1',4X,'1',4X,'2',4X,'3',4X,'4',4X,'4',
     +       4X,'5',4X,'6',4X,'7',4X,'7',4X,'8',4X,'9',2X,'REJ')
      DO 15 I1=0,9
      WRITE(22,220)I1,(IFR(I1,I2),I2=1,M+1)
  220 FORMAT(3X,I7,14I5)
   15 CONTINUE
      WRITE(22,221)CRATE
  221 FORMAT(///1X,'CORRECT CLASSIFICATION RATE = ',F8.2)
C
      WRITE(6,600)N1
  600 FORMAT(1X,'NO. OF FEATURES USED = ',I4)
      WRITE(6,221)CRATE
      WRITE(22,227)
  227 FORMAT('1')
C
C
   17 CONTINUE
C
C
      REWIND 22
      PFCMD = 'REPLACE,TAPE22='//OUTFNM//'.'
      CALL PFREQ(PFCMD,MSG,ICODE)
      IF (ICODE.NE.0) THEN
      PRINT*,MSG
      STOP
      ENDIF
C
      STOP
      END
C
  FUNCTION LOCMAX (N,NMAX,X)
C
C  THIS FUNCTION SUBPROGRAM DETERMINES THE LOCATION OF THE MAXIMUM
C  ELEMENT OF AN ARRAY
C
      DIMENSION X(NMAX)
C
      LOCMAX=1
      XMAX=X(1)
      IF (N.GT.1) THEN
      DO 1 K = 2,N
      IF (XMAX.LT.X(K)) THEN
      XMAX=X(K)
      LOCMAX = K
      ENDIF
    1 CONTINUE
      ENDIF
      END
```

APPENDIX B13

```
C
C  RECOG2:
C          PROGRAM FOR RECOGNITION OF NUMERALS USING NONPARAMETRIC
C          BAYESIAN APPROACH WITH THE ASSUMPTION OF INDEPENDENCE OF
C          FEATURES
C
   PROGRAM RECOG2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
+  INF1,INF2,INF3,INF4,INF5,TAPE11=INF1,TAPE12=INF2,TAPE13=INF3,
+  TAPE14=INF4,TAPE15=INF5,TAPE22,OUTF,TAPE20=OUTF)
C
C  INPUT FILES :
C                INF1: MIN. AND MAX. OF THE FEATURES
C                INF2: PDF'S OF CLASSES ( DIRECT ACCESS FILE )
C                INF3: FEATURE VALUES
C                INF4: FEATURES IN DESCENDING ORDER OF EFFECTIVENESS
C                INF5: SIZES OF THE CLASSES
C
C  OUTPUT FILES :
C                TAPE22: CONFUSION MATRIX ( SAVED AS 'CONFMAT' )
C                OUTF  : WRONGLY CLASSIFIED NUMERALS
C
      DIMENSION P(78,13,10), XMIN(78), XMAX(78), H(78), AP(13),
+               R(13),X(78), IFEAT(78), IFR(0:9,14), NFSIZE(40)
      DIMENSION SIZE(13), XOLD(78), XNEW(78),JOLD(78), JNEW(78)
C
      CHARACTER PFCMD*80, MSG*40
      CHARACTER OUTFNM*7
      EQUIVALENCE (X,XNEW)
C
      DATA M/13/, N/78/, NPNTS/10/, MMAX/13/, TSIZE/1000.0/
      DATA OUTFNM/'CONFMAT'/
      DATA ZEROLOG/-6.1920/
C
     WRITE(6,*)'ENTER NO. OF FEATURE SETS: NFSETS'
     READ(5,*)NFSETS
     WRITE(6,*)'ENTER SIZES OF THESE FEATURE SETS'
     READ(5,*)(NFSIZE(KK),KK=1,NFSETS)
     WRITE(6,*)'ENTER NO. OF OBS. TO BE RECOGNIZED'
     READ(5,*)NOBS
C
      REWIND 11
      REWIND 15
C
C  READ THE NO. OF OBS. IN DIFFERENT CLASSES FROM INF5
C
      READ(15,*)(SIZE(I),I=1,M)
C
C  DETERMINE THE APRIORI PROBABILITIES
C
      DO 21 I = 1,M
```

```
      AP(I) = SIZE(I)/TSIZE
   21 CONTINUE
C
C  READ THE MINIMUM AND MAXIMUM VALUES OF DIFFERENT FEATURES
C
      READ(11,110)(XMIN(K),XMAX(K),K=1,N)
  110 FORMAT(1X,8F8.4)
C
C  COMPUTE (SAMPLING INTERVAL)/2 FOR DIFFERENT FEATURES
C
      XNP=NPNTS
      DO 3 K = 1,N
      H(K) = (XMAX(K) - XMIN(K)) / (2.0*XNP)
    3 CONTINUE
C
      REWIND 22
      REWIND 20
C
C
       DO 17 KK = 1,NFSETS
C
C
C  READ THE ESTIMATED PROB. VALUES, IN DIFFERENT SAMPLING INTERVALS (J)
C  OF DIFFERENT FEATURES (K), FOR DIFFERENT CLASSES (I)
C
      REWIND 12
      DO 1 K = 1,N
      DO 2 I =1,M
      READ(12,120)(P(K,I,J),J=1,NPNTS)
  120 FORMAT(9X,10F8.4)
    2 CONTINUE
    1 CONTINUE
C
       N1 = NFSIZE(KK)
C
       REWIND 13
       REWIND 14
C
C  READ THE FEATURE NO.S TO USE FOR RECOGNITION FROM INF4
C
       READ(14,141)(IFEAT(K1),K1=1,N1)
  141 FORMAT(//////(4(16X,I4)))
C
C  WRITE THE LIST OF FEATURES ON OUTF2
C
       WRITE(22,226)N1
  226 FORMAT(1X,'NO.OF FEATURES USED = ',I4//)
       WRITE(22,225)(IFEAT(K1),K1=1,N1)
  225 FORMAT(1X,'FEATURES USED: '/(15I5))
C
C  INITIALIZE THE CELLS OF THE CONFUSION MATRIX
C
      DO 10 I1=0,9
      DO 11 I2=1,M+1
      IFR(I1,I2)=0
```

```
   11 CONTINUE
   10 CONTINUE
C
C  RECOGNITION PROCESS STARTS HERE
C
      DO 14 L = 1,NOBS
C
       READ(13,130)NUMID,INEW,(X(K),K=1,N)
  130 FORMAT(2X,I4,1X,I2/(1X,8F8.4))
C
C  MODIFY THE VALUES OF P(K,I,J)
C
      DO 22 K = 1,N
C
C  DETERMINE JNEW(K): THE SAMPLING INTERVAL CONTAINING XNEW(K)
C
      JNEW(K) = JINTV( XNEW(K),XMIN(K),H(K),NPNTS )
C
      IF (L.GT.1) THEN
      DO 23 J = 1,NPNTS
      IF (J.EQ.JOLD(K)) THEN
      P(K,IOLD,J) = (P(K,IOLD,J)*(SIZE(IOLD)-1.0)+1.0)/SIZE(IOLD)
C
      ELSE
      P(K,IOLD,J) = (P(K,IOLD,J)*(SIZE(IOLD)-1.0))/SIZE(IOLD)
      ENDIF
      IF ( P(K,IOLD,J).LT.1.0E-13 ) THEN
      P(K,IOLD,J) = 0
      ENDIF
   23 CONTINUE
      ENDIF
      DO 24 J = 1,NPNTS
      IF (J.EQ.JNEW(K)) THEN
      P(K,INEW,J) = (P(K,INEW,J)*SIZE(INEW)-1.0)/(SIZE(INEW)-1.0)
      ELSE
      P(K,INEW,J) = P(K,INEW,J)*SIZE(INEW)/(SIZE(INEW)-1.0)
      ENDIF
      IF ( P(K,INEW,J).LT.1.0E-13 ) THEN
      P(K,INEW,J) = 0
      ENDIF
   24 CONTINUE
   22 CONTINUE
C
C  STORE VALUES OF SOME VARIABLES FOR USE IN THE NEXT RECOGNITION CYCLE
C
      IOLD = INEW
      DO 25 K = 1,N
      JOLD(K) = JNEW(K)
      XOLD(K) = XNEW(K)
   25 CONTINUE
C
      ITRU = NUMID/1000
C
      DO 4 I =1,M
      R(I) = LOG(AP(I))
```

```
    4 CONTINUE
C
C  COMPUTE THE VALUES OF THE CLASSIFICATION CRITERION R(I)
      DO 5 K1 = 1,N1
C
      KVAL=IFEAT(K1)
      INTV=JNEW(KVAL)
C
C
      DO 8 I = 1,M
      IF(P(KVAL,I,INTV).LT.1.0E-13)THEN
      R(I)=R(I)+ZEROLOG
      ELSE
      R(I)=R(I)+LOG(P(KVAL,I,INTV))
      ENDIF
    8 CONTINUE
C
    5 CONTINUE
C
C  OBTAIN THE CLASS WITH MAXIMUM R-VALUE
C
      IREC = LOCMAX(M,MMAX,R)
      IF(ABS(R(IREC)).LT.1.0E-250)THEN
      IREC=14
      ENDIF
      IFR(ITRU,IREC) = IFR(ITRU,IREC) +1
C
      IDEC=ICONV(IREC)
      IF(ITRU.NE.IDEC)THEN
      WRITE(20,200)NUMID,ITRU,IDEC
  200 FORMAT(1X,3I6)
      ENDIF
C
C
    14 CONTINUE
C
C  COMPUTE THE RECOGNITION ACCURACY AND CREATE THE OUTPUT FILES
C
      CORCT=IFR(0,1)+IFR(1,2)+IFR(1,3)+IFR(2,4)+IFR(3,5)+IFR(4,6)+
+      IFR(4,7)+IFR(5,8)+IFR(6,9)+IFR(7,10)+IFR(7,11)+IFR(8,12)+
+      IFR(9,13)
      CRATE = CORCT/REAL(NOBS)*100.0
      WRITE(22,223)
  223 FORMAT(1X,'CONFUSION MATRIX :')
      WRITE(22,224)
  224 FORMAT(5X,'NUMERAL',2X,24('-'),'RECOGNIZED AS',27('-'))
      WRITE(22,222)
  222 FORMAT(14X,'0',4X,'1',4X,'1',4X,'2',4X,'3',4X,'4',4X,'4',
+      4X,'5',4X,'6',4X,'7',4X,'7',4X,'8',4X,'9',2X,'REJ')
      DO 15 I1=0,9
      WRITE(22,220)I1,(IFR(I1,I2),I2=1,M+1)
  220 FORMAT(3X,I7,14I5)
   15 CONTINUE
      WRITE(22,221)CRATE
  221 FORMAT(///1X,'CORRECT CLASSIFICATION RATE = ',F8.2)
```

```
C
      WRITE(6,600)N1
  600 FORMAT(1X,'NO. OF FEATURES USED = ',I4)
      WRITE(6,221)CRATE
      WRITE(22,227)
  227 FORMAT('1')
C
C
   17 CONTINUE
C
C
      REWIND 22
      PFCMD = 'REPLACE,TAPE22='//OUTFNM//'.'
      CALL PFREQ(PFCMD,MSG,ICODE)
      IF (ICODE.NE.0) THEN
      PRINT*,MSG
      STOP
      ENDIF
C
      STOP
      END
C
  FUNCTION LOCMAX (N,NMAX,X)
C
C  DETERMINE THE LOCATION OF THE MAXIMUM ELEMENT OF AN ARRAY
C
      DIMENSION X(NMAX)
C
      LOCMAX=1
      XMAX=X(1)
      IF (N.GT.1) THEN
      DO 1 K = 2,N
      IF (XMAX.LT.X(K)) THEN
      XMAX=X(K)
      LOCMAX = K
      ENDIF
    1 CONTINUE
      ENDIF
      ENT
C
  FUNCTION JINTV(Y,YMIN,H,NPNTS)
C
C  DETERMINES THE SAMPLING INTERVAL CONTAINING A FEATURE VALUE
C
      XNP = NPNTS
      IF(Y.LT.(YMIN+2.0*H))THEN
      JINTV = 1
      GOTO 6
      ELSEIF(Y.GE.(YMIN+2.0*(XNP-1.0)*H))THEN
      JINTV = NPNTS
      GOTO 6
      ELSE
      DO 7 J = 2,NPNTS-1
      XJ = J
      IF((Y.GE.(YMIN+2.0*(XJ-1.0)*H)).AND.(Y.LT.(YMIN+2.0*XJ*H)))THEN
```

```
      JINTV = J
      GOTO 6
      ENDIF
    7 CONTINUE
      ENDIF
    6 CONTINUE
      END
C
 FUNCTION ICONV (IREC)
C
C CONVERT SUBCLASS ID. ( 1 TO 13 ) TO CLASS ID. ( 0 TO 9 )
C
      IF(IREC.EQ.1)THEN
      ICONV=0
      ELSEIF(IREC.EQ.2.OR.IREC.EQ.3)THEN
      ICONV=1
      ELSEIF(IREC.EQ.4.OR.IREC.EQ.5)THEN
      ICONV=IREC-2
      ELSEIF(IREC.EQ.6.OR.IREQ.EQ.7)THEN
      ICONV=4
      ELSEIF(IREC.EQ.8.OR.IREC.EQ.9)THEN
      ICONV=IREC-3
      ELSEIF(IREC.EQ.10.OR.IREC.EQ.11)THEN
      ICONV=7
      ELSEIF(IREC.EQ.12.OR.IREC.EQ.13.OR.IREC.EQ.14)THEN
      ICONV=IREC-4
      ELSEIF(IREC.GE.15)THEN
      WRITE(6,609)NUMID,ITRU,IREC
  609 FORMAT(1X,3I6,'  : OCCURENCE OF IMPOSSIBLE IREC VALUE')
      STOP
      ENDIF
      RETURN
      END
```

REFERENCES

[1] S. H. Unger, 'Pattern detection and recognition,' Proc. IRE, vol. 47, pp. 1737-1752, Oct. 1959.

[2] W. G. Wee, 'A survey of pattern recognition,' IEEE Proc. Seventh Symp. on Adaptive Processes, pp. 2.e.1-2.e.13, Dec. 16-18, 1968.

[3] G. Nagy, 'State of the art in pattern recognition,' Proc. IEEE, vol. 56, pp. 836-862, May 1968.

[4] Y. C. Ho and A. K. Agrawala, 'On pattern classification algorithms - introduction and survey,' Proc. IEEE, vol. 56, pp. 2101-2114, Dec. 1968.

[5] P. M. Lewis II, 'The characteristic selection problem in recognition systems,' IRE Trans. Inform. Theory, vol. IT-8, pp. 171-178, Feb. 1962.

[6] L. A. Kamentsky and C. N. Liu, 'Computer-automated design of multifont print recognition logic,' IBM J. Research and Develop., vol. 7, pp. 2-13, Jan. 1963.

[7] C. N. Liu, 'A programmed algorithm for designing character recognition logics,' IEEE Trans. Electron. Comp., vol. EC-13, pp. 586-593, Oct. 1964.

[8] T. Marill and D. M. Green, 'On the effectiveness of receptors in recognition systems,' IRE Trans. Inform. Theory, vol. IT-9, pp. 11-17, Jan. 1963.

[9] H. Jeffreys, 'An invariant form for the prior probability in estimation problems,' Proc. Roy. Soc. A, vol. 186, pp. 453-461, 1946.

[10] S. Kullback, Information Theory and Statistics , Wiley, 1959.

[11] T. Kailath, 'The divergence and Bhattacharyya distance measures in signal selection,' IEEE Trans. Commun. Technol., vol. COM-15, pp. 52-60, Feb. 1967.

[12] A. Bhattacharyya, 'On a measure of divergence between two statistical populations defined by their probability distributions,' Bull. Calcutta Math. Soc., vol. 35, pp. 99-109, 1943.

[13] S. Watanabe, 'Karhunen-Loeve expansion and factor analysis - theoretical remarks and applications,' Proc. 4th Prague Conf. Inform. Theory, 1965.

[14] K. S. Fu, P. J. Min and T. J. Li, 'Feature selection in pattern recognition,' IEEE Trans. Syst. Sc. Cybern., vol. SSC-6, pp. 33-39, Jan. 1970.

[15] T. R. Vilmansen, 'On dependence and discrimination in pattern recognition,' IEEE Trans. Comput. (corresp.), vol. C-21, pp. 1029-1031, Sept. 1972.

[16] T. R. Vilmansen, 'Feature evaluation with measures of probabilistic dependence,' IEEE Trans. Comput., vol. C-22, pp. 381-388, Apr. 1973.

[17] L. Kanal, 'Patterns in pattern recognition: 1968-1974,' IEEE Trans. Inform. Theory, vol. IT-20, pp. 697-722, Nov. 1974.

[18] J. Kittler, 'Mathematical methods of feature selection in pattern recognition,' Int. J. Man-Machine Studies, vol. 7, pp. 609-637, 1975.

[19] C. H. Chen, 'On information and distance measures, error bounds and feature selection,' Information Sciences, vol. 10, pp. 159-173, 1976.

[20] Y. T. Chien and K. S. Fu, 'On the generalized Karhunen-Loeve expansion,' IEEE Trans. Inform. Theory, vol. IT-13, pp. 518-520, July 1967.

[21] K. Fukunaga and W. L. G. Koontz, 'Application of the Karhunen-Loeve expansion to feature selection and ordering,' vol. C-19, pp. 311-318, Apr. 1970.

[22] J. W. Sammon, 'Interactive pattern analysis and classification,' IEEE Trans. Comput., vol. C-19, pp. 594-616, July 1970.

[23] J. W. Sammon, 'An optimal discriminant plane,' IEEE Trans. Comput., vol. C-19, pp. 826-829, Sept. 1970.

[24] D. H. Foley and J. W. Sammon, 'An optimal set of discriminant vectors,' IEEE Trans. Comput., vol. C-24, pp. 281-289, Mar. 1975.

[25] J. Kittler and P. C. Young, 'A new approach to feature selection based on the Karhunen-Loeve expansion,' Pattern Recognition, vol. 5, pp. 335-352, 1973.

[26] J. Kittler, 'On the discriminant vector method of feature selection,' IEEE Trans. Comput., vol. C-26, pp. 604-606, June 1977.

[27] D. Kazakos, 'Maximin linear discrimination, I', IEEE Trans. Syst., Man and Cybern., vol. SMC-7, pp. 661-669, Sept. 1977.

[28] W. Malina, 'On an extended Fisher criterion for feature selection,' IEEE Trans. Pattern Anal. and Mach. Intel., vol. PAMI-3, pp. 611-614, Sept. 1981.

[29] H. P. Decell, P. L. Odell, and W. A. Coberly, 'Linear dimension reduction and Bayes classification,' Pattern Recognition, vol. 13, pp. 241-244, 1981.

[30] J. D. Tubbs, W. A. Coberly, and D. M. Young, 'Linear dimension reduction and Bayes classification with unknown parameters,' vol. 15, pp. 167-172, 1982.

[31] H. P. Decell and S. K. Marani, 'Feature combinations and the Bhattacharyya criterion,' Commun. in Stat. - Theory and Methods, vol. A5, pp. 1143-1152, 1976.

[32] H. P. Decell and S. M. Mayekar, 'Feature combinations and the divergence criterion,' Computers and Mathematics with Applications, vol. 3, pp. 71-76, 1977.

[33] P. L. Odell, 'A model for dimension reduction in pattern recognition using continuous data,' Pattern Recognition, Vol. 11, pp. 51-54, 1979.

[34] H. P. Decell and L. F. Guseman, 'Linear feature selection with applications,' Pattern Recognition, vol. 11, pp. 55-63, 1979.

[35] J. T. Tou and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, pp. 111-118, 1974.

[36] I. Vajda, 'Bounds of the minimal error probability on checking a finite or countable number of hypotheses,' Inf. Transmission Prob., vol. 4, pp. 9-19, 1968.

[37] I. Vajda, 'A contribution to the informational ana lysis of pattern,' in Methodologies of Pattern Recognition , S. Watanabe, Ed., New York: Academic Press, pp. 509-519, 1969.

[38] I. Vajda, 'Note on discrimination information and variation,' IEEE Trans. Inf. Theory, vol. IT-16, pp. 771-773, Nov. 1970.

[39] D. G. Lainiotis, 'A class of upper bounds on the probability of error for multihypothesis pattern recognition,' IEEE Trans. Inf. Theory, vol. IT-15, pp. 730-731, Nov. 1969.

[40] D. G. Lainiotis and S. K. Park, 'Probability of error bounds,' IEEE Trans. Syst., Man, Cybern., vol. SMC-1, pp. 175-178, Apr. 1971.

[41] C. H. Chen, 'Theoretical comparison of a class of feature selection criteria in pattern recognition,' IEEE Trans. Comput., vol. C-20, pp. 1054-1056, Sept. 1971.

[42] C. H. Chen, 'On a class of computationally efficient feature selection criteria,' Pattern Recognition, vol. 7, pp. 87-94, June 1975.

[43] G. T. Toussaint, 'Feature evaluation criteria and contextual decoding algorithms in statistical pattern recognition,' Ph.D. Dissertation , University of British Columbia, Van Couver, Canada, 1972.

[44] C. B. Chittineni, 'On the probability of error and the expected Bhattacharyya distance in multiclass pattern recognition,' Proc. IEEE, vol. 11, pp. 1451-1452, May 1972.

[45] C. B. Chittineni, 'On the application of divergence to feature selection in pattern recognition,' IEEE Trans. Syst., Man, Cybern., vol. SMC-2, pp. 668-670, Nov. 1972.

[46] C. B. Chittineni, 'On divergence and probability of error in pattern recognition,' Proc. IEEE, pp. 798-799, June 1973.

[47] C. B. Chittineni, 'On feature extraction in pattern recognition,' Internat. J. Inform. Sc., vol. 6, pp. 191-200, 1973.

[48] C. B. Chittineni, 'Efficient feature subset selection with probabilistic distance criteria,' Information Sciences, vol. 22, pp. 19-35, 1980.

[49] P. A. Devijver, 'On a new class of bounds on Bayes risk in multihypothesis pattern recognition,' IEEE Trans. Comput., vol. C-23, pp. 70-80, Jan. 1974.

[50] P. A. Devijver, 'Entropies of degree β and lower bounds for the average error rate,' Information and Control, vol. 34, pp. 222-226, July 1977.

[51] K. Matusita, 'On the theory of statistical decision functions,' Ann. Inst. Statist. Math., vol. 3, pp. 17-35, 1951.

[52] H. Chernoff, 'A measure of asymptotic efficiency for tests of a hypothesis based on a sum of observations,' Ann. Math. Stat., vol. 23, pp. 493-507, 1952.

[53] M. E. Hellman and J. Raviv, 'Probability of error, equivocation and the Chernoff bound,' IEEE Trans.Inf. Theory, vol. IT-16, pp. 368-372, July 1970.

[54] C. H. Chen, Statistical Pattern Recognition, Hayden Book Co., p.57, 1973.

[55] G. T. Toussaint, 'On the divergence between two distributions and the probability of misclassification of several decision rules,' Proc. 2nd Int. Jt. Conf. on Pattern Recognition, Copenhagen, August 1974.

[56] P. H. Swain, T. V. Robertson and A. G. Wacker, 'Comparison of the divergence and B-distance in feature selection,' LARS Information Note 020871, Purdue University, West Lafayette, Indiana, Feb. 1971.

[57] P. H. Swain and R. C. King, 'Two effective feature selection criteria for multispectral remote sensing,' LARS Information Note 042673, Purdue University, West Lafayette, Indiana, 1973.

[58] T. Lissack and K. S. Fu, 'Error estimation in pattern recognition via L^{α}-distance between posterior density functions,' IEEE Trans. Inform. Theory, vol. IT-22, pp. 34-45, Jan. 1976.

[59] E. A. Patrick and F. P. Fischer II, 'Nonparametric feature selection,' IEEE Trans. Inform. Theory, vol. IT-15, pp. 577-584, Sept. 1969.

[60] T. Ito, 'Approximate error bounds in pattern recognition,' in Machine Intelligence , Edinburgh University Press, Edinburgh, vol. 7, pp. 369-376, Nov. 1972.

[61] G. T. Toussaint, 'Distance measures as measures of certainty and their application to statistical pattern recognition,' Proc. Conf. on Theoretical and Applied Statistics and Data Analysis, Queen's University, Kingston, Ontario, June 4-6, 1973.

[62] G. T. Toussaint, 'On some measures of information and their application in pattern recognition,' Proc. Measures of Information and Their Applications, Indian Institute of Technology, Bombay, India, pp. 21-28, Aug. 16-18, 1974.

[63] I. Csiszar, 'Information-type distance measures and indirect observations,' Stud. Sci. Math., Hungar., vol. 2, pp. 299-318, 1967.

[64] I. Vajda, 'χ^{α}-divergence and generalized Fisher's information,' Transactions of the Sixth Prague Conference on Information Theory, Statistical Decision Functions Random Process, 1971, Academia, Publishing House of the Czechoslovak Academy of Sciences, Prague, 1973.

[65] K. Matusita, 'On the notion of affinity of several distributions and some of its applications, Ann. Inst. Stat. Math., vol. 19, pp. 181-192, 1967.

[66] K. Matusita, 'Some properties of affinity and applications,' Ann. Inst. Stat. Math., vol. 23, pp. 137-155, 1971.

[67] K. Matusita, 'Discrimination and the affinity of distributions,' in Discriminant Analysis and Applications, T. Cacoullos, Ed., Academic Press, pp. 213-223, 1973.

[68] G. T. Toussaint, 'Probability of error, expected divergence, and the affinity of several distributions,' IEEE Trans. Syst., Man, Cybern., vol. SMC-8, pp. 482-485, June 1978.

[69] G. T. Toussaint, 'An upper bound on the probability of misclassification in terms of the affinity,' Proc. IEEE (Letters), vol. 65, pp. 275-276, Feb. 1977.

[70] P. A. Devijver, Personal communication.

[71] G. T. Toussaint, 'Feature evaluation with a proposed generalization of Kolmogorov's variational distance and the Bhattacharyya coefficient,' Proc. Annual Canadian Conference, Montreal, pp. 422401-422413, June 1-3, 1972.

[72] C. E. Shannon, 'A mathematical theory of communication,' Bell Syst. Tech. J., vol. 27, pp. 379-423, July 1948.

[73] V. A. Kovalevsky, 'The problem of character recognition from the point of view of mathematical statistics,' in Character Readers and Pattern Recognition, V. A. Kovalevsky, Ed., Spartan Books, pp. 3-30, 1968.

[74] D. V. Lindley, 'On a measure of the information provided by an experiment,' Ann. Math. Stat., vol. 27, pp. 986-1005, 1956.

[75] H. F. Ryan, 'The information content measure as a performance criterion for feature selection,' Proc. IEEE 7th Symp. Adaptive Processes, pp. 2.c.1-2.c.11, Los Angeles, Dec. 16-18, 1968.

[76] C. W. Swonger, 'Property learning in pattern recognition systems using information content measures,' in Pattern Recognition, L. N. Kanal, Ed., Thompson, pp. 329-347, 1968.

[77] R. C. Ahlgren, H. F. Ryan, and C. W. Swonger, 'A character recognition application of an iterative procedure for feature selection,' IEEE Trans. Comput., vol. C-20, pp. 1067-1075, Sept. 1971.

[78] E. Rasek, 'A contribution to the problem of feature selection with similarity functionals in pattern recognition,' Pattern Recognition, vol. 3, pp. 31-36, Apr. 1971.

[79] D. Kozlay, 'Feature extraction in an optical character recognition machine,' IEEE Trans. Comput., vol. C-20, pp. 1063-1067, Sept. 1971.

[80] G. T. Toussaint, 'Feature evaluation with quadratic mutual information,' Information Processing Letters, vol. 1, pp. 153-156, 1972.

[81] A. Renyi, 'On measures of entropy and information,' Proc. Fourth Berkley Symp. Math. Statist. and Probab., 1960, vol. I, Univ. of California Press, pp. 547-561, 1961.

[82] G. T. Toussaint, 'A generalization of Shannon's equivocation and the Fano bound,' IEEE Trans. Syst., Man, Cybern., vol. SMC-7, pp. 300-302, Apr. 1977.

[83] P. M. Trouborst, E. Backer, D.E. Boekee, and Y. Boxma, 'New families of probabilistic distance measures,' Proc. 2nd Int. Jt. Conf. on Pattern Recognition, Copenhagen, pp. 3-5, Aug. 1974.

[84] E. Backer and A. K. Jain, 'On feature ordering in practice and some finite sample effects,' Proc. 3rd Int. Jt. Conf. on Pattern Recognition, Corolado, Calif., pp. 45-49, Nov. 1976.

[85] H. Hudimoto, 'On the distribution-free classification of an individual into one of two groups, 'Ann. Inst. Stat. Math., vol. VIII, pp. 105-112, 1956-57.

[86] H. Hudimoto, 'A note on the probability of the correct classification when the distributions are not specified,' Ann. Inst. Stat. Math., vol. IX, pp. 31-36, 1957-58.

[87] H. Jeffreys, Theory of Probability , Oxford University Press, 1948.

[88] S. Kullback and R. A. Leibler, 'On information and sufficiency,' Ann. Math. Stat., vol. 22, pp. 79-86, 1951.

[89] P. A. Devijver, Personal communication.

[90] T. T. Kadota and A. A. Shepp, 'On the best finite set of linear observables for discriminating two Gaussian signals,' IEEE Trans. Inform. Theory, vol. IT-13, pp. 278-289, Apr. 1967.

[91] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1972.

[92] T. M. Cover and P. E. Hart, 'Nearest Neighbor pattern classification,' IEEE Trans. Inform. Theory, vol. IT-13, pp. 21-27, 1967.

[93] J. T. Chu and J. C. Chueh, 'Inequalities between information measures and error probability,' J. Franklin Inst., vol. 282, pp. 121-125, Aug. 1966.

[94] D. L. Tebbe and S. J. Dwyer III, 'Uncertainty and the probability of error,' IEEE Trans. Inform. Theory, vol. IT-14, pp. 516-518, May 1968.

[95] R. M. Fano, Transmission of Information: A Statistical Theory of Communications, MIT Press, 1961.

[96] J. T. Chu and J. C. Chueh, 'Error probability in decision functions for character recognition,' J. ACM, vol. 14, pp. 273-280, Apr. 1967.

[97] G. T. Toussaint, 'Some upper bounds on error probability for multiclass pattern recognition,' IEEE Trans. Comput., vol. C-20, pp. 943-944, Aug. 1971.

[98] G. T. Toussaint, 'Some functional lower bounds on the expected divergence for multihypothesis pattern recognition, communication and radar systems,' IEEE Trans. on Syst., Man, Cybern., vol. SMC-1, pp. 384-385, Oct. 1971.

[99] L. Kanal and B. Chandrasekaran, 'On dimensionality and sample size in statistical pattern recognition,' Pattern Recognition, vol. 3, pp. 225-234, 1971.

[100] P. A. Devijver and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall International, 1982.

[101] P. C. Mahalanobis, 'On the generalized distance in Statistics,' Proc. Natl. Inst. Sc. India, vol. 12, pp. 49-55, 1936.

[102] D. J. Hand, Discrimination and Classification, John Wiley and Sons, 1981.

[103] P. A. Lachenbruch, Discriminant Analysis , Hafner Press, 1975.

[104] C. R. Rao, 'Inference on discriminant function coefficients,' in Essays in Probability and Statistics , R. C. Bose, etal. , Eds., University of North Carolina and Statistical Publishing Society, pp. 587-602, 1970.

[105] C. K. Chow, 'An optimum character recognition system using decision functions,' IRE Trans. Electron. Comput., vol. EC-6, pp. 247-254, Dec. 1957.

[106] B. Gold, 'Machine recognition of hand-sent Morse code,' IEEE Trans. Inform. Theory, vol. IT-5, pp. 17-24, Mar. 1959.

[107] W. W. Bledsoe and I. Browning, 'Pattern recognition and reading by machine,' Proc. Eastern Jt. Comput. Conf., vol. 16, pp. 225-232, 1959.

[108] R. L. Grimsdale, F. H. Sumner, C. J. Tunis, and T. Kilburn, 'A system for the automatic recognition of patterns,' Proc. IEE, vol. 106, Part B, pp. 210-221, 1959.

[109] L. D. Harmon, 'Automatic recognition of print and script,' Proc. IEEE, vol. 60, pp. 1165-1176, Oct. 1972.

[110] C. Y. Suen, M. Berthod, and S. Mori, 'Automatic recognition of handprinted characters - the state of the art, 'Proc. IEEE, vol. 68, pp. 469-487, Apr. 1980.

[111] G. Nagy, 'Optical character recognition - theory and practice,' in Handbook of Statistics , vol. 2, P. R. Krishnaiah and L. N. Kanal, Eds., North Holland, pp. 621-649, 1982.

[112] J. R. Ullmann, Pattern Recognition Techniques , London Butterworths, 1973.

[113] S. H. Unger, 'A computer oriented toward spatial problems,' Proc. IRE, vol. 46, pp. 1744-1750, Oct. 1958.

[114] C. Y. Suen, 'Distinctive features in automatic recognition of handprinted characters,' Signal Processing, vol. 4, pp. 193-207, Apr. 1982.

[115] R. Bakis, N. M. Herbst, and G. Nagy, 'An experimental study of machine recognition of handprinted numerals,' IEEE Trans. on Syst., Sc., Cybern., vol. SSC-4, pp. 119-132, July 1968.

[116] J. R. Ullmann, 'Experiments with the n-tuple method of pattern recognition,' IEEE Trans. Comput., vol. 18, pp. 1135-1137, Dec. 1969.

[117] A. A. Spanjersberg, 'Combinations of different systems for the recognition of handwritten digits,' Proc. 2nd Int. Jt. Conf. on Pattern Recognition, pp. 208-209, Aug. 1974.

[118] N. D. Tucker and F. C. Evans, 'A two-step strategy for character recognition using geometrical moments,' Proc. 2nd Int. Jt. Conf. on Pattern Recognition, pp. 223-225, Aug. 1974.

[119] A. B. S. Hussain, G. T. Toussaint, and R. W. Donaldson, 'Results obtained using a simple character recognition procedure on Munson's handprinted data,' IEEE Trans. Comput., vol. 21, pp. 201-205, Feb. 1972.

[120] H. A. Glucksman, 'Classification of mixed-font alphabetics by characteristic loci,' in Digest of 1st Ann. IEEE Comput. Conf., pp. 138-141, Sept. 1967.

[121] H. A. Glucksman, 'Multicategory classification of patterns represented by high-order vectors of multilevel measurements,' IEEE Trans. Comput., vol. 20, pp. 1593-1598, Dec. 1971.

[122] A. L. Knoll, 'Experiments with characteristic loci for recognition of handprinted characters,' IEEE Trans. Comput., vol. EC-18, pp. 366-372, Apr. 1969.

[123] M. Michael and W. C. Lin, 'Experimental study of information measure and inter-intra class distance ratios on feature selection and orderings,' IEEE Trans. Syst., Man, Cybern., vol. SMC-3, pp. 172-181, Mar. 1973.

[124] P. Krause, W. Schwerdtman, and D. Paul, 'Two modifications of a recognition system with pattern series expansion and Bayes classifier,' Proc. 2nd Int. Jt. Conf. on Pattern Recognition, pp. 215-219, Aug. 1974.

[125] A. Gudesen, 'Quantitative analysis of preprocessing techniques for the recognition of handprinted characters,' Pattern Recognition, vol. 8, pp. 219-227, 1976.

[126] H. Niemann, 'Classification of characters by man and machine,' Pattern Recognition, vol. 9, pp. 173-179, 1977.

[127] G. H. Granlund, 'Fourier preprocessing for handprint character recognition,' IEEE Trans. Comput., vol. 21, pp. 195-201, 1972.

[128] E. Persoon and K. S. Fu, 'Shape discrimination using Fourier descriptors,' IEEE Trans. Syst., Man, Cybern., vol. 7, pp. 170-179, 1977.

[129] H. C. Andrews, 'Multidimensional rotations in feature selection,' IEEE Trans. Comput., vol. 20, pp. 1045-1051, 1971.

[130] H. Genechi, K. I. Mori, S. Watanabe, and S. Katsuragi, 'Recognition of handwritten numeral characters for automatic letter sorting,' Proc. IEEE, vol. 56, pp. 1292-1301, 1968.

[131] D. M. Stern and D. W. C. Shen, 'Character recognition by context-dependent transformations,' Proc. IEE, vol. 11, pp. 1923-1932, 1964.

[132] J. H. Munson, 'Experiments in the recognition of handprinted text: Part I - Character recognition,' Proc. FJCC, vol. 33, pp. 1125-1138, 1968.

[133] R. J. Spinrad, 'Machine recognition of handprinting,' Inform. Control, vol. 8, pp. 124-142, 1965.

[134] F. Ali and T. Pavlidis, 'Syntactic recognition of handwritten numerals,' IEEE Trans. Syst., Man, Cybern., vol. SMC-7, pp. 537-541, 1977.

[135] T. Pavlidis and F. Ali, 'Computer recognition of handwritten numerals by polygonal approximation,' IEEE Trans. Syst., Man, Cybern., vol. 5, pp. 610-614, 1975.

[136] D. J. Quarmby and J. Rastall, 'Experiments on handwritten numeral classification,' IEEE Trans. Syst, Man, Cybern., vol. 1, pp. 331-338, 1971.

[137] K. Yamamoto and S. Mori, 'Recognition of handprinted characters by an outermost point method,' Pattern Recognition, vol. 12, pp. 229-236, 1980.

[138] G. T. Toussaint and R. W. Donaldson, 'Algorithms for recognizing contour-traced handprinted characters,' IEEE Trans. Comput., vol. 19, pp. 541-546, 1970.

[139] R. Narasimhan and V. S. N. Reddy, 'A syntax-aided recognition scheme for handprinted English letters,' Pattern Recognition, vol. 3, pp. 346-361, 1971.

[140] J. T. Tou and R. C. Gonzalez, 'Recognition of handwritten characters by topological feature extraction and multilevel categorization,' IEEE Trans. Comput., vol. 19, pp. 776-785, 1972.

[141] K. S. Fu, Syntactic Pattern Recognition and Applications , Prentice-Hall International, 1982.

[142] B. Duerr, W. Haettich, H. Tropf, and G. Winkler, 'A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals,' Pattern Recognition, vol. 12, pp. 189-199, 1980.

[143] T. R. Vilmansen, 'Information and distance measures with application to feature evaluation and to heuristic sequential classification,' PhD Thesis , University of British Columbia, Vancouver, Canada, 1974.

[144] C. C. Kwan, L. Pang, and C. Y. Suen, 'A comparative study of some recognition algorithms in character recognition,' Proc. Int. Conf. Cybern. Soc., pp. 530-535, 1979.

[145] E. Parzen, 'On estimation of probability density function and mode,' Ann. Math. Stat., vol. 33, pp. 1065-1076, 1962.

[146] V. K. Murthy, 'Estimation of probability density,' Ann. Math. Stat., vol. 36, pp. 1027-1031, 1965.

[147] I. J. Good, The Esimation of Probabilities: An Essay on Modern Bayesian Methods, Researh Monograph 30, Cambridge, Mass: MIT Press, 1965.

[148] V. K. Murthy, 'Nonparametric estimation of multivariate densities with applications,' in Multivariate Analysis , P. R. Krishnaiah, Ed., pp. 43-56, New York: Academic Press, 1966.

[149] T. J. Wagner, 'Nonparametric estimates of probability densities,' IEEE Trans. Inform. Theory, vol. IT-21, pp. 438-440, July 1975.

[150] S. D. Morgera, 'Structural estimation - multivariate probability density estimation,' in Pattern Recognition in Practice, Gelsema and Kanal, Eds., pp. 369-380, 1980.

[151] C. R. Rao and S. K. Mitra, Generalized Inverse of Matrices and Its Applications, New York: Wiley, 1971.

[152] J. H. Wilkinson and C. Reinsch, Handbook for Automatic Computation, vol. II: Linear Algebra , Springer-Verlag, 1971.

[153] M. G. Kendall, Rank Correlation Methods , London: Charles Griffin and Company, 1948.

[154] P. A. Lachenbruch and M. R. Mickey, 'Estimation of error rates in discriminant analysis,' Technometrics, vol. 10, pp. 1-11, Feb 1968.