Purdue University
Purdue e-Pubs

ECE Technical Reports

Electrical and Computer Engineering

2-1-1993

DESIGN OF PARTIALLY SUPERVISED CLASSIFIERS FOR MULTISPECTRAL IMAGE DATA

Byeungwoo Jeon
Purdue University School of Electrical Engineering

David Landgrebe
Purdue University School of Electrical Engineering

DESIGN OF
PARTIALLY SUPERVISED CLASSIFIERS
FOR MULTISPECTRAL IMAGE DATA

Byeungwoo Jeon
David Landgrebe

TR-EE 93-11
February, 1993

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285

This work was sponsored in part by NASA under Grant NAGW-925

TABLE OF CONTENTS

ABSTRACT

This report addresses a partially supervised classification problem, especially when the class definition and corresponding training samples are provided a *priori* only for just one particular class. In practical applications of pattern classification techniques, a frequently observed characteristic is the heavy, often nearly impossible requirements on representative prior statistical class characteristics of all classes in a given data set. Considering the effort in both time and man-power required to have a well-defined, exhaustive list of classes with a corresponding representative set of training samples, this "partially" supervised capability would be very desirable, assuming adequate classifier performance can be obtained.

Two different classification algorithms are developed to achieve simplicity in classifier design by reducing the requirement of prior statistical information without sacrificing significant classifying capability. The first one is based on optimal significance testing, where the optimal acceptance probability is estimated directly from the data set.

In the second approach, the partially supervised classification is considered as a problem of unsupervised clustering with initially one known cluster or class. A weighted unsupervised clustering procedure is developed to automatically define other classes and estimate their class statistics.

The operational simplicity thus realized should makes these partially supervised classification schemes very viable tools in pattern classification.

CHAPTER 1

INTRODUCTION

1.1 Information and Pattern Classification in Remote Sensing

For decades, the technology of remote sensing has been successfully applied in many interdisciplinary applications of Earth observational data. Pattern classification methods have had a major role in applying remote sensing technology. A pattern classification system can be described generally as in following schematic.

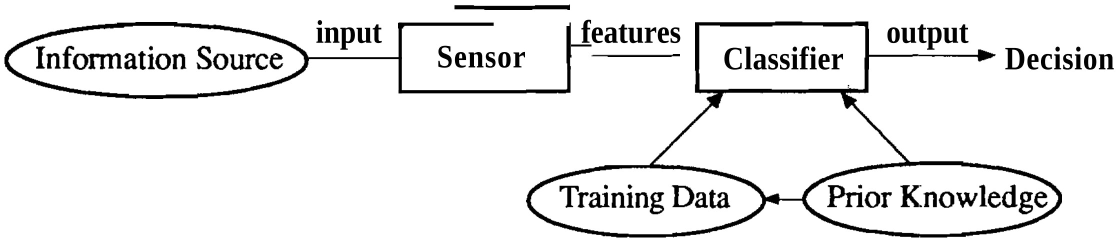


Figure 1.1 General Schematic of Pattern Classification.

The incoming information-bearing data are analyzed and classified into one of the pre-defined categories. To have a proper classification of given data, one needs to decide what classifier to employ and which features to use in the classification. A well-defined, informative, and exhaustive list of classes, and a representative set of training samples from which the statistical characteristics of all classes can be estimated is essential.

If prior knowledge about the statistical characteristics of the categories or classes is available, usually in terms of training samples, the classifier is referred to as "supervised." The major portion of prior knowledge is often in the form of training samples with known class labels. In this case, 'the class statistics are estimated from the available set of labeled training samples. When there is no prior knowledge, then, the classifier is referred to as "unsupervised." In many cases, the training samples are available only for a subset of classes, or, training samples are gathered only for those particular classes. Considering the expensive process of gathering training samples in both man-power and time, this situation is not uncommon in practice especially when one needs to identify only a subset of classes. It can be referred as a "partially supervised" classification problem.

1.2 Design of Partially Supervised Classifiers

In practical applications of pattern classification techniques, it is not unusual to confront a task in which only a particular subset of classes, for which training samples are available, are desired to be recognized or identified. **A** design of conventional **supervised** classifier requires training samples for all the classes in the given data in order to perform optimally. Considering the effort in both time and man-power required to have a well-defined, exhaustive list of classes with a corresponding representative set of training samples, this "partially" supervised capability would be very desirable, assuming adequate classifier performance can be obtained. This report addresses the partially supervised classification problem, especially when the class definition and corresponding training samples are provided a *priori* only for just one particular class.

Two different approaches are investigated. The first one is based on optimal significance testing, where the optimal acceptance probability is estimated directly from the data set. In the second approach, the partially supervised classification is considered as a problem of unsupervised clustering with initially one known cluster or class. The definitions and statistics of the other classes are automatically developed through a weighted unsupervised clustering procedure which is developed to keep the cluster corresponding to the "class of interest" from losing its identity as the "class of interest." Once all the classes are developed, a conventional relative classifier such as a maximum likelihood classifier is used in the classification.

Even though the partially supervised classification algorithms are to perform at best comparable to one in which all the classes and statistical characteristics are available, considering the time and effort required for collecting ground truth, or training samples required for defining all the existent classes in the given data set, this will be very useful in practice when a data-analyst is interested in identifying only samples belonging to a certain class.

1.3 Organization of the Report

The outline of this report is as follows.

Chapter 2 addresses an absolute classification approach based on the optimal significance testing where the optimal accept probability is estimated from the given data set without user's supervision.

In Chapter 3, the problem of partially supervised classification is formulated as that of a relative classification with only one a *priori* known class. Weighted unsupervised clustering algorithm is investigated for unsupervised development of class definition and statistical characteristics necessary succeeding relative classification. Following the experimental results are conclusions and suggestions for further research regarding the design problem of partially supervised classifiers.

CHAPTER 2

PARTIALLY SUPERVISED CLASSIFICATION WITH OPTIMAL SIGNIFICANCE TESTING

2.1 Introduction

Successful classification of given data sets requires a proper design of classifiers to be employed. The design or training of classifiers is performed using prior information which is usually gathered in the form of training samples. The number of training samples necessary is dependent on the number of features and the number of classes. Generally, the process of obtaining training samples is very expensive in terms of both time and manpower. In practical applications of pattern classification techniques, a frequently observed characteristic is the heavy, often nearly impossible requirements on representative prior statistical class characteristics of all classes in a given data set.

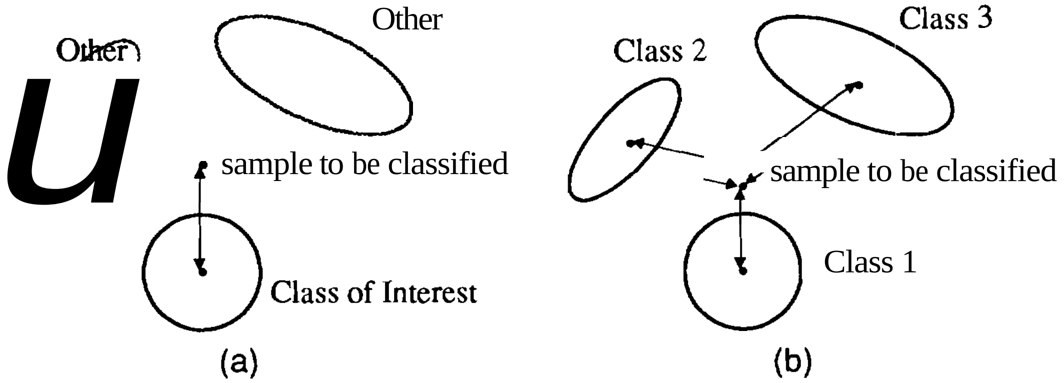


Figure **2.1** Two Different Classification Schemes. (a) Absolute Classification Scheme. (b) Relative Classification Scheme.

Broadly speaking, classification analysis schemes can be dichotomized into two different categories, one being based on an absolute classification scheme and the other based on a relative classification scheme. Classifiers based on the absolute scheme, such as a parallelepiped classifier (Richards 86), or a scheme based upon a known absorption feature for a specific material, classify data samples on an absolute basis, *i.e.*, without regard to the spectral responses of other materials or classes which may be in the scene. In such cases, class definition through training samples is required only for the particular class under consideration. There may be many applications where one wants to recognize only a single class of pixels. For example, one might be interested in finding only the pixels belonging to a class, "corn," etc. This absolute classification scheme is very attractive in this case.

The scheme in the second category is "relative classification" where class decisions are made on a relative basis. The maximum likelihood classifier, one of the most widely used relative classifiers, assigns a pixel to the class which has the largest likelihood value relative to other classes. Therefore, even if only one class is of interest, training samples must be obtained for all other classes also to adequately train the classifier. The necessity of supplying training samples for, or otherwise defining all other classes can be an onerous shortcoming especially when there are large numbers of classes and/or features to deal with. While a properly designed relative classifier can nearly always provide better performance, and is very much less sensitive to many unmanageable factors, e.g., atmospheric conditions, calibration, etc., the operational simplicity of the absolute scheme may make it the scheme! of choice in many instances.

This report addresses the design problem of partially supervised classifiers, especially when the class definition and corresponding training samples are provided a *priori* for only one particular class as in the absolute classification schemes. Two different approaches are investigated. The first one is based on *optimal* significance testing. The investigation of this approach addresses the problem of estimating, without supervision by the data-analyst, an optimal significance level, or equivalently, an optimal acceptance probability, which is an indispensable element in significance testing.

In the second approach which will be introduced in the next Chapter, the advantages of both a reduced requirement on obtaining training samples in the absolute classification and the potentially robust and powerful discriminating capability of a relative classifier are sought by developing an automatic mechanism of extracting statistical information corresponding to an "others" class without recourse to the training samples supplied by anallyst. That is, the classification algorithms proposed can develop class definitions and corresponding class statistics, requiring the user to supply prior knowledge only pertaining to the particular class under consideration.

The organization of this Chapter is as follows. After a brief introduction in Section 2.2 on a partially supervised classification approach based on significance testing, Section 2.3 address on an optimal significance testing procedure where an estimating algorithm of an optimal acceptisnce probability with a given optimality criterion is presented. Section 2.4 shows the experimental results of this optimal significance testing in the context of the partially supervised classification problem.

2.2 Partially Supervised Classification with Significance Testing

Significance testing is a widely used technique in various applications of statistical analysis, such as classification, or object detection (Therrien et *al.* 86). It is especially useful in such problems as the single hypothesis problem (Fukunaga et *al.* 87, Quatieri 83) where one is to identify a particular class of objects among others with only statistical information pertaining to those objects of interest. This kind of problem can arise when defining all tlhe classes and gathering corresponding statistical information is impossible, or very expensive in terms of time and manpower. Examples of applying significance testing techniques include target detection, object detection out of various backgrounds (Quatieri 83), texture detection, cloud detection, fault or anomaly detection in diagnostic monitoring (Bello 92).

Significance testing can be used for partially supervised classification when there is only one class of interest and the class definition and it class statistics are available a *priori* only for that class. Note that significance testing is based on the absolute classification scheme in Fig. 2.1.(a).

One of the important elements in significance testing is the acceptance probability (or, significance level) which must be provided by the data analyst usually in such a way that the type I (*i.e.*, omission) error rate is kept within a pre-specified level (Drake 67). Obviously, omission error is not necessarily the only relevant criterion to consider in determining a suitable acceptance probability, and there are many other possible optimality conditions. For instance, the acceptance probability could be selected on the basis of the Bayes minimum error criterion. The criteria used in the minimax test, or Neyman-Pearson test (Van Trees 68) might also be used in selecting the desired acceptance probability.

Unfortunately, lack of prior statistical information other than that of the particular class of interest may prevent directly applying conventional procedures used in hypothesis testing. The commission error, or type II error can not be easily evaluated unless the relative distribution of all classes in the given data set is available. Note that a mixture density estimates of the feature vectors can give an estimate of the probability density of the "others" class if the prior probability of the class of interest is known. For significance testing, requiring only some appropriate measure of the distances of samples from the mean of the class of interest, it suffices to estimate a one dimensional mixture probability density of the distances, not the multidimensional features vectors.

In the following is presented an algorithm which can automatically estimate the optimal acceptance probability from the given data set under the selected optimality conditions, such as Bayes total classification error, minimum class-averaged classification error, or the generalized total classification error criteria. With this estimated optimal acceptance probability, classification can be performed to identify the class of interest.

This automatic estimation of the proper acceptance probability will be doubtlessly desirable, at least to the user with little prior knowledge about the data set. The algorithm to be proposed in this chapter can also be applied to the case where the specific class of interest consists of several sub-classes. When there are a large number of constituent sub-classes of the given class of interest and the sample distributions of the sub-classes are quite different from each other, this automatic estimation capability should be very handy, since one doesn't need to undertake the manual selecting process of an acceptance probability for each sub-class.

2.3 Optimal Significance Testing

Suppose there is a data set, $X \equiv \{x_1, ---, x_N\}$ with N samples. Each data point, x_i, is a q-dimensional feature vector ($q \geq 1$). It is assumed that one is only interested in identifying a single class which is denoted by C_{int}, *i.e.*, discriminating between it and the "others" class, denoted by C_{others}. The "others" class might consist of several classes none of which is one's interest. Prior statistical knowledge is assumed to be available only for the class of interest. Let $f_X(x \mid C_{int})$ and $f_X(x \mid C_{others})$ be the probability density functions of classes C_{int} and C_{others}, respectively. The prior probabilities of C_{int} and C_{others} are indicated by π_{int} and π_{others}. It is assumed either to know the density function $f_X(x \mid C_{int})$, or, to have a set of representative training samples of C_{int} from which a reasonably accurate estimate of $f_X(x \mid C_{int})$ can be made. In general, $f_X(x \mid C_{others})$, π_{int} and π_{others} are not known other than the fact that $\pi_{int} + \pi_{others} = 1$. The mixture probability density, denoted as $f_X(x)$, is written as,

$$f_X(x) = \pi_{int}\, f_X(x \mid C_{int}) + \pi_{others}\, f_X(x \mid C_{others}) \qquad (2.1)$$
$$\text{where, } 0 < \pi_{int}, \pi_{others} < 1,\ \pi_{int} + \pi_{others} = 1$$

Even though the following derivations do not require any specific family of probability density functions for C_{int}, multivariate normality will be assumed for C_{int} for simplicity's sake. Generalization to other probability density functions is straightforward. Furthermore, without loss of generality, C_{int} can be assumed to have a zero mean, denoted by $\mathbf{O}_q$, and an identity covariance matrix, denoted by $\mathbf{I}_{qxq}$. This standard multivariate normal distribution will be denoted by $\mathrm{MVN}[\mathbf{O}_q, \mathbf{I}_{qxq}]$.

In significance testing, a single hypothesis $H_1 : x \in C_{int}$, is tested against all other alternatives. The degree of support for the hypothesis H_1 is measured with *test statistic,* $T(x)$ which is a function of feature vector x, $x \in X$. With $f_X(x \mid C_{int})$ being $\mathrm{MVN}[\mathbf{O}_q, \mathbf{I}_{qxq}]$, a natural choice for the test statistic would be $T(x) = x^T x$ by which the significance test rejects sample x if $T(x) > \lambda$. Once the test statistic is selected, the threshold λ specifies the ***rejection region*** in the feature space.

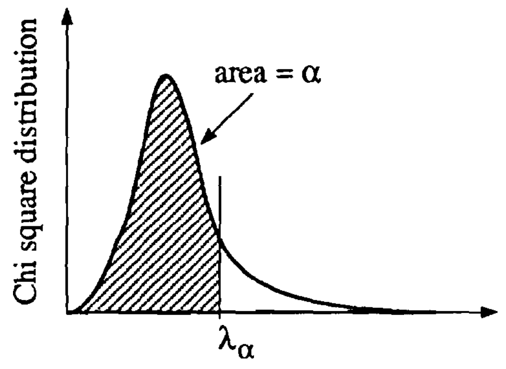


Figure 2.2 Threshold λ_α Corresponding to a Significance Level (1 - a).; $T(x) = x^T x$ and $f_x(x \mid C_{int})$ is $MVN[O_q, I_{qxq}]$.

Choosing an appropriate rejection region (or equivalently, the threshold λ) is an important problem which deserves further attention. The availability of the necessary statistical characterization of C_{int} enables control of the omission error, denoted by ε_1, using,

$$\varepsilon_1 = P\{T(x) > \lambda_\alpha \mid H_1\} \leq 1 - \alpha , \qquad 0 \leq \alpha \leq 1 \qquad (2.2.a)$$

The value, (1 - a) defines the maximum allowable omission error and is often called the ***significance*** level or ***rejection*** probability. The parameter a will be called the ***acceptance probability.*** The threshold associated with α, denoted by λ_α, can be obtained by solving,

$$\int_0^{\lambda_\alpha} f_Y(y \mid C_{int}) \, dy = \alpha \qquad (2.2.b)$$

where $f_Y(y \mid C_{int})$ is the conditional distribution of $y = T(x) = x^T x$, under the hypothesis H_1. (The notation of H_1 and C_{int} will be used interchangeably). When $f_x(x \mid C_{int})$ is $MVN[O_q, I_{qxq}]$, $f_Y(y \mid C_{int})$ is known to be the chi-squared distribution with q degrees of freedom.

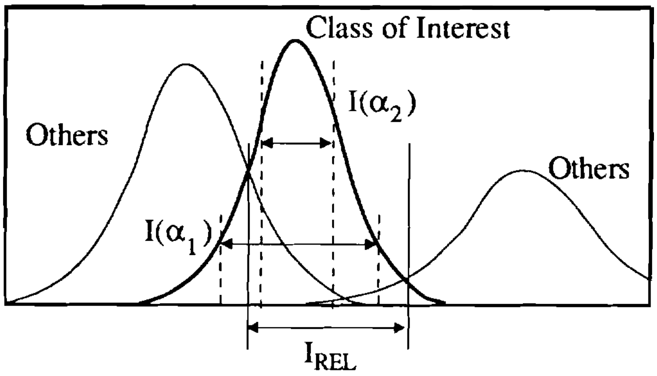


Figure 2.3 Decision Regions of the Class of Interest with Significance Testing.; An improper significance level may result in either an excessive omission or commission error.;„„I - the decision region of a relative classifier, such as a maximum likelihood classifier.; $I(\alpha_1)$, $I(\alpha_2)$ - the decision regions of significance testing with levels (1-a,), (1-a,), respectively.

While the omission error ε_1 can be controlled within a certain value specified by (1-a) through eq. (2.2.a,b), the commission error, denoted by ę is generally very difficult to control, as discussed before, since its evaluation requires frequently unavailable statistical knowledge about all alternatives. By increasing the acceptance probability a, the omission error can be reduced, but, at the same time, the commission will be increased.

The omission error plotted versus the acceptance probability has a slope of –1, but the slope of commission error is dependent on the closeness of the distribution of the "others" class to the "class of interest." To avoid potentially excessive omission or commission errors, the acceptance probability a must be carefully determined by checking the relative distribution of data samples with respect to the class of interest. An automatic estimation capability of optimum acceptance probability is thus very desirable.

Since the estimation problem of optimum acceptance probability will be addressed in a similar fashion to the hypothesis testing, a brief review of a simple binary hypothesis testing procedure (Van Trees 68) is worthwhile. Assume a simple hypothesis test with two hypothesis H_1 and H_2 from which one is to be selected. The Bayesian framework requires two sets of parameters, *i.e.*, one set including prior probabilities associated with the hypothesis and the other set with associated costs. Each cost is associated with the corresponding course of action as in Fig. 2.4.

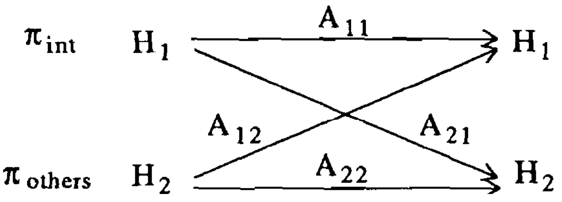


Figure 2.4 Prior Probabilities (π_{int}, π_{others}) and Costs A_{ij}'s in a Binary Hypothesis Test.; A_{ij} refers to the cost given to accepting hypothesis H_i when H_j is true.

A_{ij} is the cost given to the action of accepting hypothesis H_i when H_j is true. It is quite logical to set $A_{11} = A_{22} = 0$, that is, no cost is assigned to a correct decision. Without loss of generality, the other costs can be set as $A_{21} = A \cdot A_{12}$ with proper A, A > 0, where A_{12} doesn't affect the design of the optimal test and thus, can be dropped out in the average cost function. The optimal test can be designed by minimizing the a ***posteriori*** expected cost given as,

$$E_{Bayes} = A\ \pi_{int}\ \varepsilon_1 + \pi_{others}\ \varepsilon_2 \tag{2.3}$$

ε_1 and ε_2 are the omission and commission error probability, respectively and computed using $f_x(x \mid H_i)$, the probability density function of x under H_i, $i = 1, 2$, as,

$$\varepsilon_1 = \int_{Z_2} f_x(x \mid H_1)\ dx$$

$$\varepsilon_2 = \int_{Z_1} f_x(x \mid H_2)\ dx$$

where, Z_i is the decision region for H_i, $i = 1, 2$

Note that if (A, π_{int}, π_{others}) are known, then, an optimal Bayes minimum expected cost test can be designed. It is well known that this test is the likelihood ratio test (LRT) whose design requires selection of an appropriate threshold, based upon parameters (A, π_{int}, π_{others}), which, in turn, requires knowledge of ε_1 and ε_2 as functions of the threshold. In significance testing which can be viewed as a problem of ***single*** hypothesis testing, the optimal acceptance probability can be obtained in a similar way to the simple binary hypothesis counterpart. That is, instead of ε_1 and ε_2 as functions of the threshold, they can be obtained as functions of the acceptance probability. Unfortunately, the expression for commission error ε_2 in significance testing, is ordinarily not readily available a priori, since the probability density function under hypothesis H_2 is not known. Nevertheless, estimating the commission error function for a given data set is possible, as will be discussed in following section. With the estimated ε_2, the same idea of simple binary hypothesis testing mentioned above can be applied also to significance testing in estimating the optimal acceptance probability.

2.4 Estimation of Optimal Acceptance Probability

In this section, an algorithm which can automatically estimate the optimal acceptance probability by checking the actual relative data distributions is presented. There can be many different optimality criteria for the acceptance probability. For example, the acceptance probability can be selected solely on the basis of the omission error or commission error, or, it can be selected based on a criterion which is basically a weighted sum of omission and commission errors. In this section, three different optimality conditions are considered in selecting a proper acceptance probability.

2.4.1 Omission and Commission Errors as Functions of Acceptance Probability a

Suppose there are N_1 samples belonging to C_{int} in the data set X. N_1 is unknown, in general. Then, in the data set X, there will be $N_2 \equiv N - N_1$, data points from the class C_{others}. Assume the prior probabilities are,

$$\pi_{int} = \frac{N_1}{N} \quad \text{and} \quad \pi_{others} = \frac{N_2}{N}$$

The expected number of data points in X accepted with the acceptance probability a is denoted as $N(\alpha)$ and written as a function of a, $0 \leq a \leq 1$, as,

$$N(\alpha) \equiv N \int_0^{\lambda_\alpha} f_Y(s)\, ds \tag{2.4}$$

where $f_Y(y)$ is the mixture probability density function of y, $y = x^T x$, $y \geq 0$, and λ_α is the threshold corresponding to the acceptance probability a in eq. (2.2.b). $f_Y(y \mid C_{int})$ is similarly defined as a probability density function of $y = x'x$, $x \in C_{int}$. $N(\alpha)$ is a monotonically increasing function of a in the interval 0 ■ a ■ 1, since,

$$\frac{d\lambda_\alpha}{d\alpha} = \frac{1}{f_Y\,(\lambda_\alpha \mid C_{int})} > 0 \tag{2.5.a}$$

and,

$$\frac{dN(\alpha)}{d\alpha} = N \cdot f_Y\,(\lambda_\alpha) \cdot \left[\frac{d\lambda_\alpha}{d\alpha}\right] \geq 0, \qquad 0 \leq \alpha \leq 1 \tag{2.5.b}$$

Although the mixture density $f_Y\,(y)$, is not available a *priori*, it can be easily estimated using the y values where $y = x^T x$, $x \in X$. In a similar way, $N_1(\alpha)$ and $N_2(\alpha)$, the expected numbers of data points accepted with acceptance probability a, $0 \leq a \leq 1$, and coming from C_{int} and C_{others}, respectively, are written as,

$$N_1(\alpha) \equiv N_1 \int_0^{\lambda_\alpha} f_Y(s|C_{int})\, ds \tag{2.6.a}$$

$$N_2(\alpha) \equiv N_2 \int_0^{\lambda_\alpha} f_Y(s|C_{others})\, ds \tag{2.6.b}$$

$f_Y(y|C_{others})$ is the density function of y's corresponding to C_{others}. $N_1(a)$ and $N_2(\alpha)$ are also monotonically increasing in $0 \leq a \leq 1$. From the relations in eq. (2.1) and eq. (2.2.b), $N_1(\alpha)$ and $N_2(\alpha)$ are simplified as,

$$N_1(\alpha) = \alpha \cdot N_1 \tag{2.7.a}$$

$$N_2(\alpha) = N(\alpha) - N_1(\alpha) = N(\alpha) - \alpha \cdot N_1 \tag{2.7.b}$$

Using eq. (2.5.b) and eq. (2.7.b), an upper bound of prior probability π_{int} can be obtained as,

$$\pi_{int} \leq \min_{0 \, \mathrm{I} \, a \, \mathrm{I} \, 1} \left[\frac{f_Y(y)}{f_Y(y|C_{int})} \right]_{y = \lambda_\alpha} \tag{2.8}$$

Now, compute the omission and commission errors at acceptance probability a. The omission error rate, denoted by $\varepsilon_1(\alpha)$, is obtained by dividing the number of C_{int} samples rejected at acceptance probability a with N_1.

$$\varepsilon_1(\alpha) = \frac{N_1 - N_1(\alpha)}{N_1} = 1 - \alpha \tag{2.9}$$

Similarly, the commission error rate, denoted by $\varepsilon_2(\alpha)$, is obtained by dividing the number of accepted C_{others} samples by N_2, with given acceptance probability a.

$$\varepsilon_2(\alpha) = \frac{N(\alpha) - \alpha \bullet N_1}{N - N_1} \tag{2.10}$$

Note that, with respect to a, $\varepsilon_1(\alpha)$ is a strictly decreasing function with slope -1 and $\varepsilon_2(\alpha)$ is a monotonically increasing function, but the actual rate of increase of $\varepsilon_2(\alpha)$ is dependent on the behavior of $N(\alpha)$. The evaluation of $\varepsilon_2(\alpha)$ generally requires knowledge of N_1, or equivalently, the prior probability π_{int}.

The optimal acceptance probability a is dependent on the criterion which assesses the optimality. In many situations, a simple average of omission and commission errors,

$$E_1(\alpha) \equiv \frac{1}{2} \left[\varepsilon_1(\alpha) + \varepsilon_2(\alpha) \right] \tag{2.11.a}$$

serves as a good candidate for assessing optimality. Under the Bayesian total probability error criterion, the optimal acceptance probability minimizes,

$$E_2(\alpha) \equiv \pi_{int} \, \varepsilon_1(\alpha) + \pi_{others} \, \varepsilon_2(\alpha) \tag{2.11.b}$$

the sum of omission and commission errors weighted with the prior probabilities. This weighting can be generalized by allowing different cost (or, risk) between omission and commission errors as,

$$E_3(\alpha) \equiv A \bullet \pi_{int} \, \varepsilon_1(\alpha) + \pi_{others} \, \varepsilon_2(\alpha) \tag{2.11.c}$$

Constant A, where A > 0, is the risk or cost on making ***omission*** errors relative to the risk of making ***commission*** error being 1. The criteria in eq. (2.11.a,b) can be considered to be special cases of $E_3(\alpha)$. That is, $E_3(\alpha)$ with A = 1, is the same as $E_2(\alpha)$. Setting $A = \pi_{others}/\pi_{int}$ makes $E_3(\alpha)$ equivalent to $E_1(a)$. The criterion in eq. (2.11.c) will be called the "generalized" total classification error criterion.

Note that identifying a specific single class, or detecting specific objects from the background in a given scene can be considered as a two class classification problem and a confusion matrix can be drawn as in Table 2.1. (N_{11} is a number of C_{int} samples correctly classified as C_{int} and N_{22} is a number of C_{others} samples correctly classified as C_{others}).

Table 2.1 **Confusion Matrix.**

		Assigned class	
		C_{int}	C_{others}
Actual	C_{int}	N_{11}	$N_1 - N_{11}$
Class	C_{others}	$N_2 - N_{22}$	N_{22}

Then the classification error **probabilities** of C_{int} and C_{others} are equivalent to the omission and commission errors, respectively.

$$\text{error probabilities of } C_{int} = 1 - \frac{N_{11}}{N_1}$$

$$\text{error probabilities of } C_{others} = 1 - \frac{N_{22}}{N_2}$$

Two criteria have been conventionally used in assessing classification performance. One is the "overall classification error" which is computed as the ratio of the total number of errors to the total number of samples in the given data set. The other is the "class-averaged classification error," and it is a simple average of the classwise classification errors. Notice that the overall classification error is no more than a weighted sum of the classwise classification errors according to the prior probabilities. Thus, it is equivalent to the total probability of error in eq. (2.1l.b). By the way, the "class-averaged classification error" is equivalent to eq. (2.11.a). The class-averaged classification error criterion is a very useful indicator of classification performance especially when there are large differences between prior probabilities since the overall classification accuracy will be dominated by the performance of the class having the dominant prior probability. In applying significance testing, there will be many cases when the number of data points belonging to one class is dominantly large than the others. In these cases, the class-averaged classification error in eq. (2.11.a) will be desirable in assessing optimality.

2.4.2 Estimating Optimum Acceptance Probability

In following discussion, only the criterion in eq. (2.11.c) will be used since the others can be derived as special cases of this criterion by setting an appropriate value of A. The optimal acceptance probability a can be obtained by minimizing $E_3(\alpha)$ with respect to a over the interval, $0 \leq a \leq 1$. That is, by equating the first order derivative of $E_3(\alpha)$ to 0,

$$\begin{aligned}\frac{dE_3(\alpha)}{d\alpha} &= \frac{1}{N}\left[N_2\frac{d\varepsilon_2(\alpha)}{d\alpha} - A \cdot N_1\right] \\ &= \frac{1}{N}\left[\frac{dN(\alpha)}{d\alpha} - (1+A) \cdot N_1\right] = 0\end{aligned} \qquad (2.12.a)$$

and checking the sign of the second order derivative in eq. (2.12.b) below, the optimal value of a which gives the minimum value of $E_3(\alpha)$ can be found. Note that solving eq. (2.12.a) requires, in general, knowledge of N_1, or, equivalently, the prior probability π_{int}.

$$\frac{d^2E_3(\alpha)}{d\alpha^2} = \pi_{others}\frac{d^2\varepsilon_2(\alpha)}{d\alpha^2} = \frac{1}{N}\frac{d^2N(\alpha)}{d\alpha^2} \qquad (2.12.b)$$

Since the second derivative of $\varepsilon_1(\alpha)$ is zero, eq. (2.12.b) is only affected by the commission error, $\varepsilon_2(\alpha)$. Substituting the first order derivative of $N(\alpha)$ given in eq. (2.5.b) into eq. (2.12.a) results in,

$$N\, f_Y(\lambda_\alpha) = (1+A)\, N_1\, f_Y(\lambda_\alpha | C_{int}) \qquad (2.13)$$

The first order derivative of $E_3(\alpha)$ being always positive in $0 \le a \le 1$ indicates that $Nf_Y(\lambda_\alpha)$ on the left side of eq. (2.13) is always larger than the right side, $(1+A)N_1 f_Y(\lambda_\alpha|C_{int})$ for all a in the interval [0,1]. Since (1+A) > 1, this means that the data points expected to be in the infinitesimal region $(\lambda_\alpha, \lambda_\alpha + d\lambda_\alpha)$ are always more than the expected number of C_{int} samples in the region and thus, considerable commission error will result no matter how restrictive the acceptance probability is. Therefore, the optimum value of a is expected to be 0. On the other hands, the first order derivative of $E_3(\alpha)$, being always negative in the interval, $0 \le a \le 1$, indicates, in the same token, that the data points expected to be in the infinitesimal region $(\lambda_\alpha, \lambda_\alpha + d\lambda_\alpha)$ are always less than the expected number of C_{int} samples in the region (which is weighted by (1+A)), therefore, the possibility of commission error is very low. This will allow acceptance probability a to increase up to 1.

Since increasing a would not only decrease the omission error but also increase simultaneously the commission error, other than these two extreme cases, minimum points of $E_3(\alpha)$ will be located where the degree of increase in the weighted commission error starts to surpass the decrease of the weighted omission error. The prior probabilities and relative cost A determine the actual balancing between omission and commission errors. Due to the closed interval of a, [0,1], the minimum of $E_3(\alpha)$ always exists and so does the optimum a, even if there may be no solution satisfying eq. (2.12.b) and the positivity of eq. (2.12.c). Suppose solutions satisfying these two conditions do exist, and denote a set of those solutions as S.

$$S \equiv \{\, \alpha \mid \frac{dE_3(\alpha)}{da} = 0 \;\text{ and }\; \frac{d^2E_3(\alpha)}{da^2} > 0,\; 0 \le \alpha \le 1 \}$$

Then, each element in Swill correspond to a (local) minimum of $E_3(\alpha)$. The global minimum can be selected by comparing the actual values of $E_3(\alpha)$ at different a's in S in the following way. Suppose α_i and α_j are elements in S, then, the difference, $E_3(\alpha_i) - E_3(\alpha_j)$ can be written as,

$$E_3(\alpha_i) - E_3(\alpha_j) = \frac{\Delta_{ii}}{N} \qquad (2.14)$$

$$\text{where, } \Delta_{ij} \equiv \left[N(\alpha_i) - N(\alpha_j) - (\alpha_i - \alpha_j) \cdot (1+A) \cdot N_1 \right]$$

By checking the signs of the Δ_{ij}'s, the acceptance probability which attains a global minimum of $E_3(\alpha)$ can be selected from the set S. Notice that evaluating eq. (2.14) requires the prior probability, π_{int}, but, in the case of the class-averaged classification error criterion, it can be evaluated even without knowing π_{int} since substituting $A = \pi_{others}/\pi_{int} = N_2/N_1$ results in a quantity independent of π_{int} as,

$$(1+A)\, N_1 = (1 + \frac{N_2}{N_1})\, N_1 = N$$

This property of the class-averaged classification error criterion will be very useful in actual application of this algorithm since the prior probabilities are unknown in most problems.

2.4.3 Optimum Acceptance Probabilities for the Sub-classes of the Class of Interest

Frequently, one has a class of interest which consists of several sub-classes. These sub-classes are components of the original class which is often referred to as an "information class" (Swain 78). The term "information class" implies a physically meaningful entity. One cannot always model the statistical distribution of the given information class with a known simple distribution function. In this case, the information class can be decomposed into several sub-classes, each of which is described with a simple known probability density function, such as the Gaussian distribution function. This decomposition of the information class into a set of sub-classes can be accomplished using clustering in the feature space (Swain 78). These sub-classes generally might not correspond to any physically meaningful entity, since they are selected to describe the data distributions of the information class in the feature space. When there are several sub-classes belonging to one information class, significance testing can be performed in following manner.

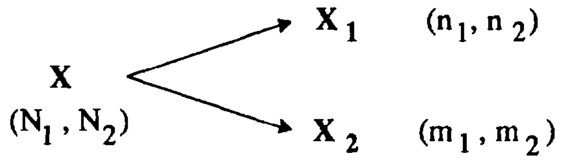


Figure 2.5 Division of Data Set X into Two Subsets, X_1 and X_2.; n_1 and m_1 are the numbers of class-of-interest samples respectively in the subsets X_1 and X_2. Similarly, n_2 and m_2 are the numbers of samples from the class "others," found respectively in the subsets X_1 and X_2.

Suppose there are two sub-classes. Since the statistical characteristics of the two sub-classes are assumed available, the given data set X can be divided into two subsets, one for "sub-class 1" and the other for "sub-class 2" as in Fig. 2.4 by applying any classifier, for example, a maximum likelihood classifier.

Significance testing is applied to each subset to obtain samples which should be accepted, and the union of samples accepted from each subset is the result of significance testing applied to the given information class. In this approach, the optimal acceptance probability is selected separately for each sub-class according to the relative distributions of samples in the corresponding subset. The estimating capability of the optimal acceptance probabilities for each sub-class will certainly be useful when there are a large number of sub-classes and the relative distributions of samples in each sub-class are quite different from each other, since one doesn't need to undertake the manual selection process of proper acceptance probability for each sub-class.

The optimality of the estimated acceptance probabilities can be assessed either at the sub-class level, or at the information class level. If the acceptance probabilities are selected to achieve the given optimality independently in each sub-class, then they are said to be optimal at the "sub-class level." On the other hand, the acceptance probabilities are called optimal at the "information class level" if they attain the given optimality for the union of accepted samples from the sub-classes. The acceptance probabilities optimal at the sub-class level do not necessarily retain the same optimality at the information class level.

Suppose α_1 and α_2, the optimal acceptance probabilities respectively for sub-class 1 and 2, are to be estimated employing the generalized total probability of error criterion of eq. (2.11.c), written as $E_3(\alpha_1,\alpha_2)$, at the information class level.

$$E_3(\alpha_1,\alpha_2) = A \cdot \pi_{int}\, \varepsilon_1(\alpha_1,\alpha_2) + \pi_{others}\, \varepsilon_2(\alpha_1,\alpha_2) \qquad (2.15)$$

where $\varepsilon_1(\alpha_1,\alpha_2)$ is the omission error of the given information class with α_1 and α_2 for sub-classes 1 and 2. Similarly, $\varepsilon_2(\alpha_1,\alpha_2)$ is the corresponding commission error. Omission error $\varepsilon_1(\alpha_1,\alpha_2)$ has two components. One is $\varepsilon_1^1(\alpha_1)$, the omission error occurring in sub-class 1 and the other, $\varepsilon_1^2(\alpha_2)$, the same omission error occurring in sub-class 2. Likewise, the commission error $\varepsilon_2(\alpha_1,\alpha_2)$ can be computed with two components, $\varepsilon_2^1(\alpha_1)$ and $\varepsilon_2^2(\alpha_2)$, commission errors occurring respectively in sub-class 1 and 2.

$$\varepsilon_1(\alpha_1, \alpha_2) = \frac{n_1\, \varepsilon_1^1(\alpha_1) + m_1\, \varepsilon_1^2(\alpha_2)}{N_1} \qquad (2.16.a)$$

$$\varepsilon_2(\alpha_1, \alpha_2) = \frac{n_2\, \varepsilon_2^1(\alpha_1) + m_2\, \varepsilon_2^2(\alpha_2)}{N_2} \qquad (2.16.b)$$

After a few algebraic operations, $E_3(\alpha_1,\alpha_2)$ in eq. (2.15) can be written in terms of the criterion in eq. (2.11.c) evaluated at each sub-class as,

$$E_3(\alpha_1,\alpha_2) = \frac{n_1+n_2}{N} E_3(\alpha_1) + \frac{m_1+m_2}{N} E_3(\alpha_2) \qquad (2.17.a)$$

where,

$$E_3(\alpha_1) = A \cdot \pi_1^1\, \varepsilon_1^1(\alpha_1) + \pi_2^1\, \varepsilon_2^1(\alpha_1) \qquad (2.17.b)$$

$$E_3(\alpha_2) = A \cdot \pi_1^2\, \varepsilon_1^2(\alpha_2) + \pi_2^2\, \varepsilon_2^2(\alpha_2) \qquad (2.17.c)$$

$$\pi_1^1 = \frac{n_1}{n_1+n_2} \quad \text{and } \pi_2^1 = 1 - \pi_1^1$$

$$\pi_1^2 = \frac{m_1}{m_1+m_2} \quad \text{and } \pi_2^2 = 1 - \pi_1^2$$

Note that minimization of eq. (2.17.a) can be achieved by minimizing $E_3(\alpha_1)$ and $E_3(\alpha_2)$ given in eq. (2.17.b,c) ***independently.*** Therefore, in the case of the (generalized) total classification error criteria in eq. (2.11.b,c), estimating the optimal acceptance probability independently for each sub-class at the sub-class level always leads to the same optimality also at the information class level. Hence, there is no inconsistency in the optimality for those cases. The result in eq. (2.17.a) is also applicable to the class-averaged classification error criterion if the relative weight A is substituted by π_{others}/π_{int}. As seen in $E_3(\alpha_1)$ and $E_3(\alpha_2)$ in eq. (2.17.b,c), this substitution of the A value doesn't lead to the same class-averaged classification error criterion in sub-class 1, 2, unless the following two equations are satisfied.

$$A \cdot \pi_1^1 = \pi_2^1 \quad \text{and} \quad A \cdot \pi_1^2 = \pi_2^2 \qquad (2.18.a)$$

These two equations above can be satisfied if the following relation holds.

$$\frac{n_1}{n_2} = \frac{m_1}{m_2} \qquad (2.18.b)$$

Therefore, unless eq. (2.18.b) is satisfied, optimality in the sense of the class-averaged classification error criterion at the information class level cannot be achieved by applying the same criterion to each sub-class. However, optimality based on the class-averaged classification error criterion can be accomplished at the level of the information class if the generalized total classification error criterion with A satisfying eq. (2.18.a) is used in each sub-class.

2.4.4 Probability Density Function Estimation

In computing an optimum acceptance probability a, density estimation is required to compute $N(\alpha)$ in eq. (2.4). Since $N(\alpha)$ is the expected number of samples accepted with acceptance probability a, it can be obtained, in the most simplistic way, by counting the number of samples whose test statistic is less than the threshold λ_α while varying the acceptance probability a. The first order derivative of $N(\alpha)$ is then obtained by numerical differentiation of $N(\alpha)$. Even though this method is quite simple and fast enough, it has some drawbacks. For example, the counting nature in estimation causes discontinuities in $N(\alpha)$ and consequently, brings difficulties in calculating the derivative. Furthermore, different ways of discretizing the interval [0,1] of a in counting samples can produce different estimates of $N(\alpha)$. This is similar to the problem of histogram-based density estimation where the estimated density can vary depending on bin definition (Silverman 86). Due to these considerations, the proposed algorithm uses a kernel-based Parzen density estimate which has been not only rigorously studied but also has been widely applied in many fields of application. If the kernel function is denoted as K(•), then, the probability density estimate $f_Y(s)$, $s \geq 0$ can be written as,

$$f_Y(s) = \frac{1}{Nh} \sum_{Y} K\left[\frac{s - y}{h}\right] \qquad (2.19)$$

$$\text{where, } \int_{-\infty}^{+\infty} K(s)ds = 1$$

The summation in eq. (2.19) is carried out for all y's, $y = x^T x$, $x \in X$ and N is the total number of data points in X. The variable h in eq. (2.19), is called the window size (or, smoothing parameter) of the kernel function. This determines how much smoothing is allowed in estimating the density. Selecting an appropriate window size h can be cumbersome sometimes since an improper window size h can result in either under-smoothing, or over-smoothing which might cause some degree of uncertainty in locating the optimal acceptance probability. It is possible to compute an optimal window size which is dependent on the kernel function, dimensionality and the number of samples N (p.86 in (Silverman 86)).

Since the values of y are all non-negative, the domain of the density estimation is $[0,+\infty)$. In this case, the use of a symmetric kernel function such as the Gaussian kernel function will result in underestimation near zero since there are no samples in the negative region. This underestimation can be avoided by using positive reflection techniques (Boneva et *al.* 71) in which a new density estimate is obtained with an augmented set of y's. Suppose $f_Y^*(s)$ is the density estimate acquired with the augmented data set. Then, the desired density estimate, $f_Y(s)$, in the region of $s \geq 0$, is obtained by doubling the density estimate acquired with the augmented data set as,

$$\begin{array}{ll} f_Y(s) = 2\, f_Y^*(s) & \text{if } s \geq 0 \\ f_Y(s) = 0 & \text{otherwise} \end{array}$$

The augmented data set is obtained by including the reflected values of y's against the origin 0 additionally in the original set of y values.

2.5 Experiments and Discussion

To test the performance of the proposed estimating algorithm for optimal acceptance probability in significance testing, experiments were carried out with both simulated and real data. In the case of simulated data, several bivariate Gaussian data sets were generated to simulate data sets with a wide range of separability. In the case of real data, Landsat Thematic Mapper (TM) data were used. For the optimality assessment, the class-averaged classification error and the total classification error criterion were used.

2.5.1 Experiment with Simulated Data

For a test with simulated data, 1000 samples were generated for the class of interest to be bivariate Gaussian (*i.e.*, the dimensionality 9 = 2) with zero mean and an identity covariance matrix. For the class "others," 2000 samples were generated to be bivariate Gaussian with a mean $[d,0]^T$, $d > 0$, and an identity covariance matrix.

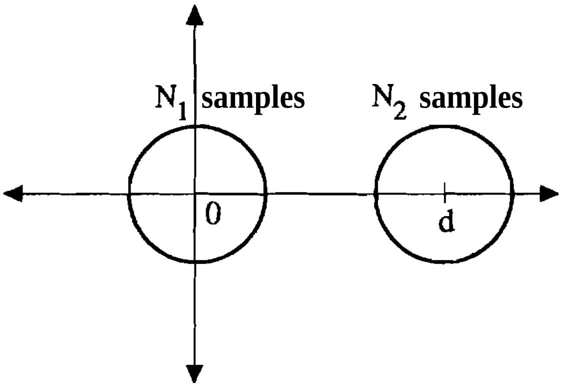


Figure 2.6 Simulated 2 Class, 2 Dimensional Gaussian Data Sets.; C_{int} : 1000 samples in $MVN[0_q, I_{qxq}]$, C_{others} : 2000 samples in $MVN[[d,0]^T, I_{qxq}]$, $(N_1 = 1000, N_2 = 2000, q=2)$.

With this set-up, the exact amount of overlap between the two distributions can be calculated. The term "overlap" is defined here as the volume which is shared by the two probability density functions. That is, when the distance between two classes is d, the overlap between the two classes is given as,

$$\text{Overlap}(d) = 1 - \frac{2}{\sqrt{2\pi}} \int_0^{d/2} \exp(-\frac{1}{2} s^2)\ ds$$

By varying d, the distance between the two class means, data sets with different degrees of overlap can be simulated. d was increased from 0.1 to 5 in steps of 0.1. If d = 0.1, there is 96.02% of overlap between the two distributions, and in the case of d=5, there is only 1.24% of overlap. To avoid any random error due to the data generation process and its effect on evaluating the experimental result, data sets were generated 50 times with different seed numbers, and the averaged result was used in comparison.

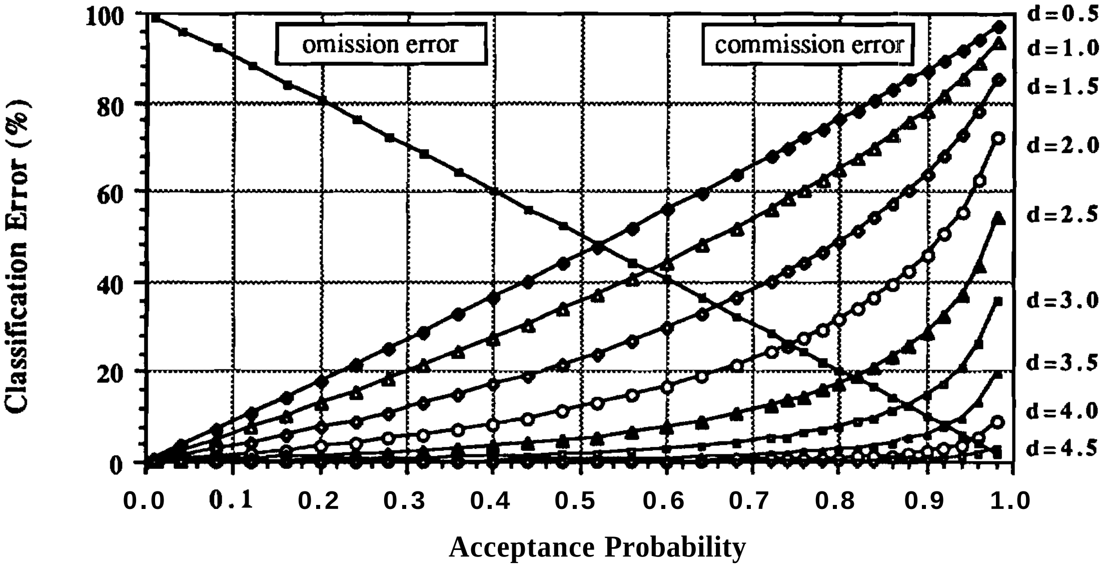


Figure 2.7 Omission and Commission Errors with Respect to Acceptance Probability.; d is the distance between two class means.

At first, various different acceptance probability a's were examined in significance testing by increasing a from 0.01 to 0.99 in steps of 0.01 to see its dependence on a as shown in Fig. 2.7. As expected, the omission error decreased linearly with respect to the acceptance probability with slope = -1. In the case of commission error, the slope of increase depended on the degree of overlap between the two distributions. When d = 0.5 which resulted in 80.26% of overlap between the two distributions, the commission error increased almost linearly with respect to a. This is due to the substantial closeness of the two distributions. When there was effectively no overlap such as in the case d = 4.5

(2.44% of overlap), the commission error stayed very low, virtually insensitive to a. The resulting class-averaged error in eq. (2.11.a) and the total probability error in eq. (2.11.b) are shown in Fig. 2.8 and 2.9.

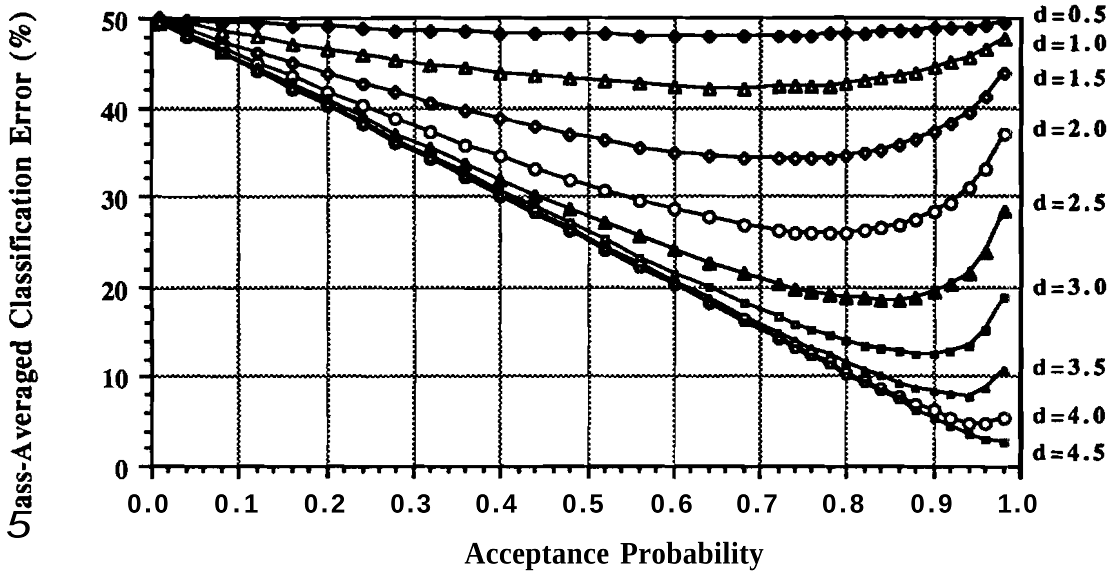


Figure 2.8 Class-Averaged Error versus Acceptance Probability a.; d is the distance between the two class means.

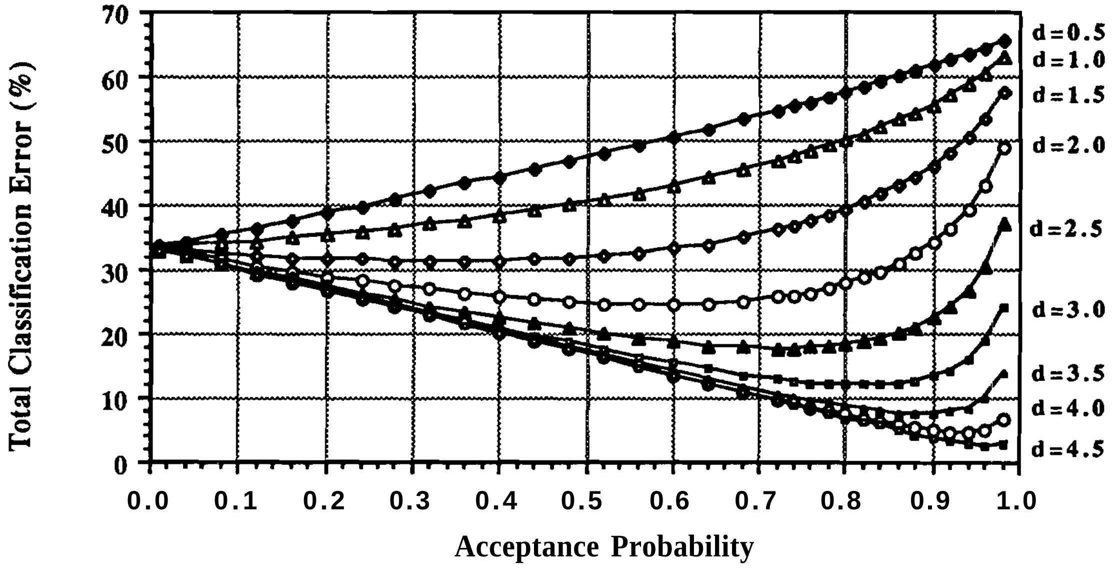


Figure 2.9 Total Classification Error versus Acceptance Probability a.; d is the distance between the two class means.

In the cases of d=0.5 and d=1.0, the total probability error and the class-averaged error had very gentle slopes.

To make a comparison with the estimated values, optimal acceptance probabilities were manually selected by changing a from 0.01 to 0.99 in steps of 0.01 and choosing the best one based on the selected optimality criterion. These manually selected were denoted by "scanned" values and compared with the estimates obtained by the proposed algorithm.

The estimated acceptance probabilities with both the class-averaged and the total classification error criteria are shown in Fig. 2.10. When applying the total classification error criterion, the true value of prior probabilities were used.

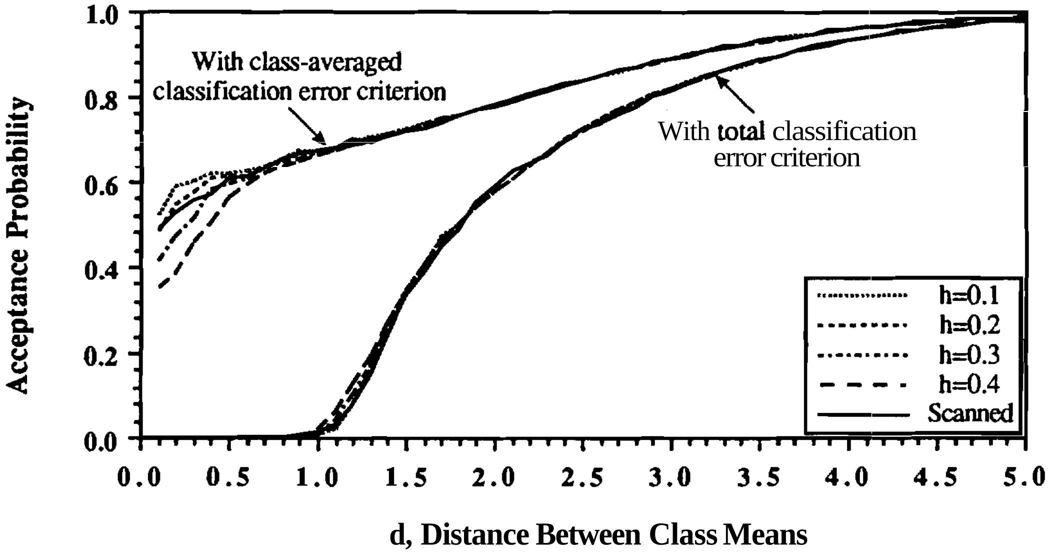


Figure 2.10 **Estimated Optimal Acceptance Probability versus d, the Distance Between Two Class Means.; Solid lines show the manually selected acceptance probabilities. Dotted lines show the estimated optimal acceptance probabilities using the proposed method. h is window size.**

The density estimate required for N(α) was obtained employing a Gaussian Kernel-based parzen density estimate with the data set augmented by positive reflection (Boneva et *al.* 71). Even though an appropriate kernel window size h was computed as 0.2 based on (Silverman 86), several different values were also tested to see its effect on the estimated acceptance probabilities. In Fig. 2.10, the estimated values followed very closely those manually selected especially when the distance d was large. The optimal acceptance probability based on the total classification error criterion was near 0 when d was not large enough, since the total classification error was an increasing function of acceptance probability for those small d values as seen in Fig. 2.9. For example, when d < 1.0, the number of commission errors increases almost at the same rate as that by which omissions decreased due to the significant amount of overlap between the two class distributions as seen in Fig. 2.7. Because the prior probability of C_{int} is less than that of C_{others}, the omission error is weighted less than the commission error under the total classification error criterion. This explains why the acceptance probabilities when d < 1.0, were very small under the total classification error criterion. When d < 1.0 with the class-averaged classification error criterion, some degree of difference was observed between the estimated and the manually selected value. Since the curve of class-averaged classification error was nearly flat when d < 1.0 as seen in Fig. 2.8, an exact location of the minimum of the class-averaged classification error was hard to pinpoint and thus, there was a relatively large standard deviation not only in the estimated but also in the manually selected optimum a values as shown in Fig. 2.11.

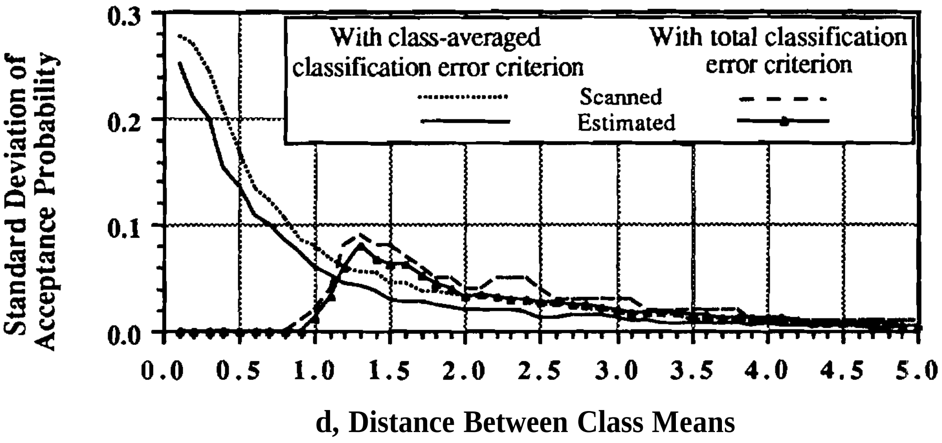


Figure 2.11 Standard Deviation of Optimal Acceptance Probabilities versus the Distance Between the Two Class Means.; Window size h=0.2.

In spite of those discrepancies in estimated a values when d < 1.0, there was not much difference in the resulting class-averaged classification errors. Note that the slope of the total classification error curve in Fig. 2.9 was nearly zero in the lower acceptance probability region. For the same reason, in Fig. 2.11, there was observed also relatively large deviations under the total classification error criterion in the region $1.0 < d < 2.0$.

Since less than 1% of difference in classification errors were observed under both optimality error criteria by varying the window size, classification results are shown only for h=0.2 in following Fig. 2.12 and 2.13.

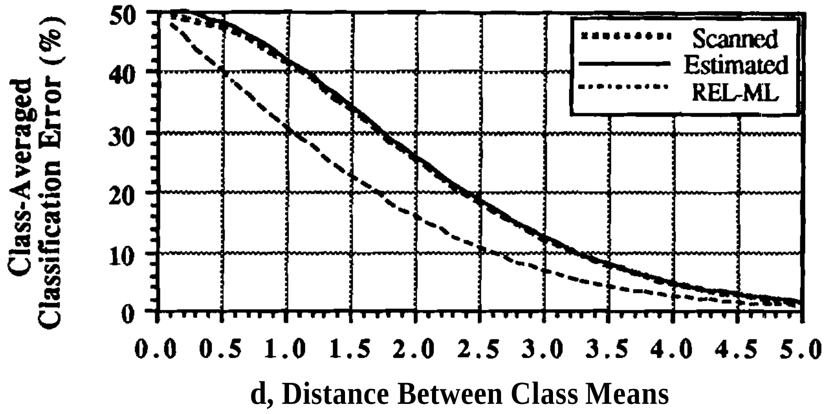


Figure 2.12 Class-Averaged Classification Error versus the Distance Between the Two Class Means.; Acceptance probabilities were estimated with the class-averaged classification error criterion. "REL-ML" is a result with the relative maximum likelihood classifier. Window size h = 0.2.

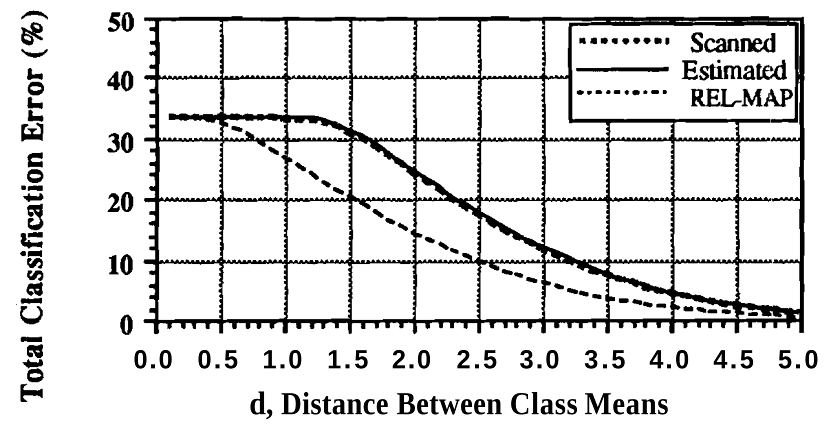


Figure 2.13 Total Classification Error versus the Distance Between the Two Class Means.; Acceptance probabilities are estimated with the total classification error criterion. "REL-MAP" is a result with the relative maximum *a posteriori* classifier. Window size h = 0.2.

The significance test deals with only the values of the selected test statistic, therefore there is a dimensionality reduction of feature vectors to one-dimensional space of the selected test statistic, and this causes loss of valuable information in classification. To see the effect of dimensionality reduction, a (relative) maximum likelihood classifier (denoted as "REL-ML") and a maximum a ***posteriori*** classifier (denoted as "REL-MAP") were designed in the original q-dimensional space with known class statistics of C_{int} and C_{others}. Their classification results were also included in Fig. 2.12 and 2.13 to see the effect of dimension reduction. Under both optimality conditions, the estimated optimal acceptance probabilities resulted in almost the same performances with manually selected values. There was a maximum of about 12% error increase due to the dimensionality reduction.

To see the effect of the data reflection on estimating optimal acceptance probabilities, the same experiment was performed, but without data reflection. Density estimation without reflected data would be expected to introduce underestimation of the probability density $f_Y(y)$ near y = 0 due to using a symmetric kernel function with only positive y values. This underestimation in $f_Y(y)$ and subsequently in $N(\alpha)$ near y = 0 would cause underestimation of commission errors, therefore, the optimal acceptance probability estimates would be expected to be larger than they should be. Since the Gaussian kernel function rapidly decreases as its argument becomes larger, this effect of under-estimation would exist only in the region near y=0. Figure 2.14 shows the differences in estimated acceptance probabilities, computed as, $\alpha_{\text{without positive reflection}} - \alpha_{\text{with positive reflection}}$.

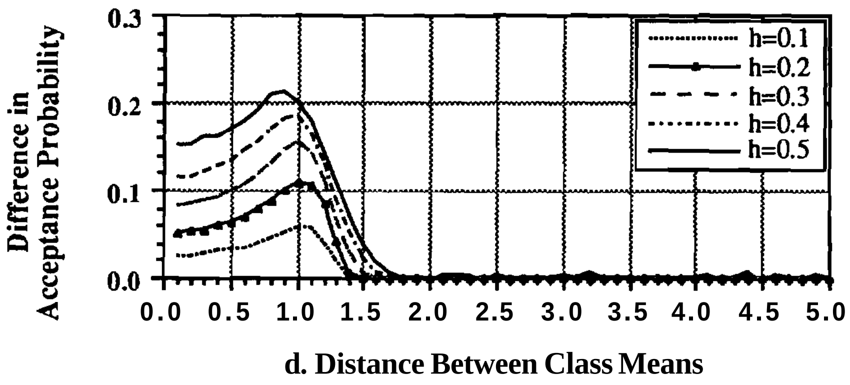


Figure 2.14 Differences in Acceptance Probabilities with and without Data Reflection under the Total Classification Error Criterion.; h is the kernel window size.

No difference was observed with the class-averaged classification error criterion. However, there were differences in the case of the total classification error criterion. As seen in Fig. 2.14, the estimated optimal acceptance probabilities without data reflection were larger by as much as 0.2 compared to those with data reflection in the region of d < 1.5. However, there was no significant difference when d > 1.5. Greater differences were observed as the window size h became larger, since the large window size would have more reflected samples in the summation of the kernel function values. The reflection technique in estimating a probability density function is observed to be necessary if the acceptance probabilities are expected to be near zero.

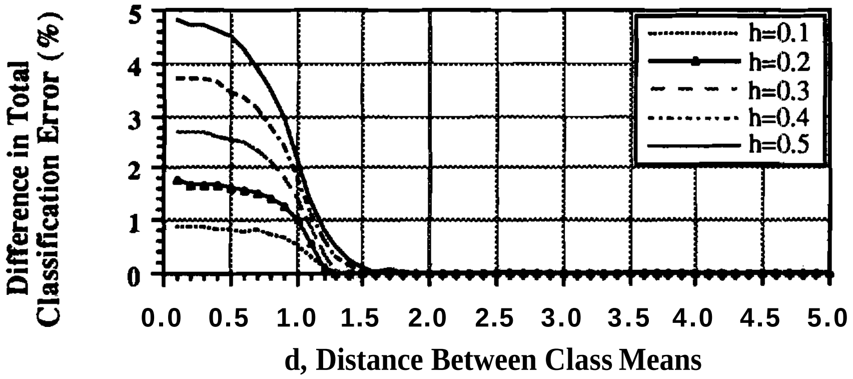


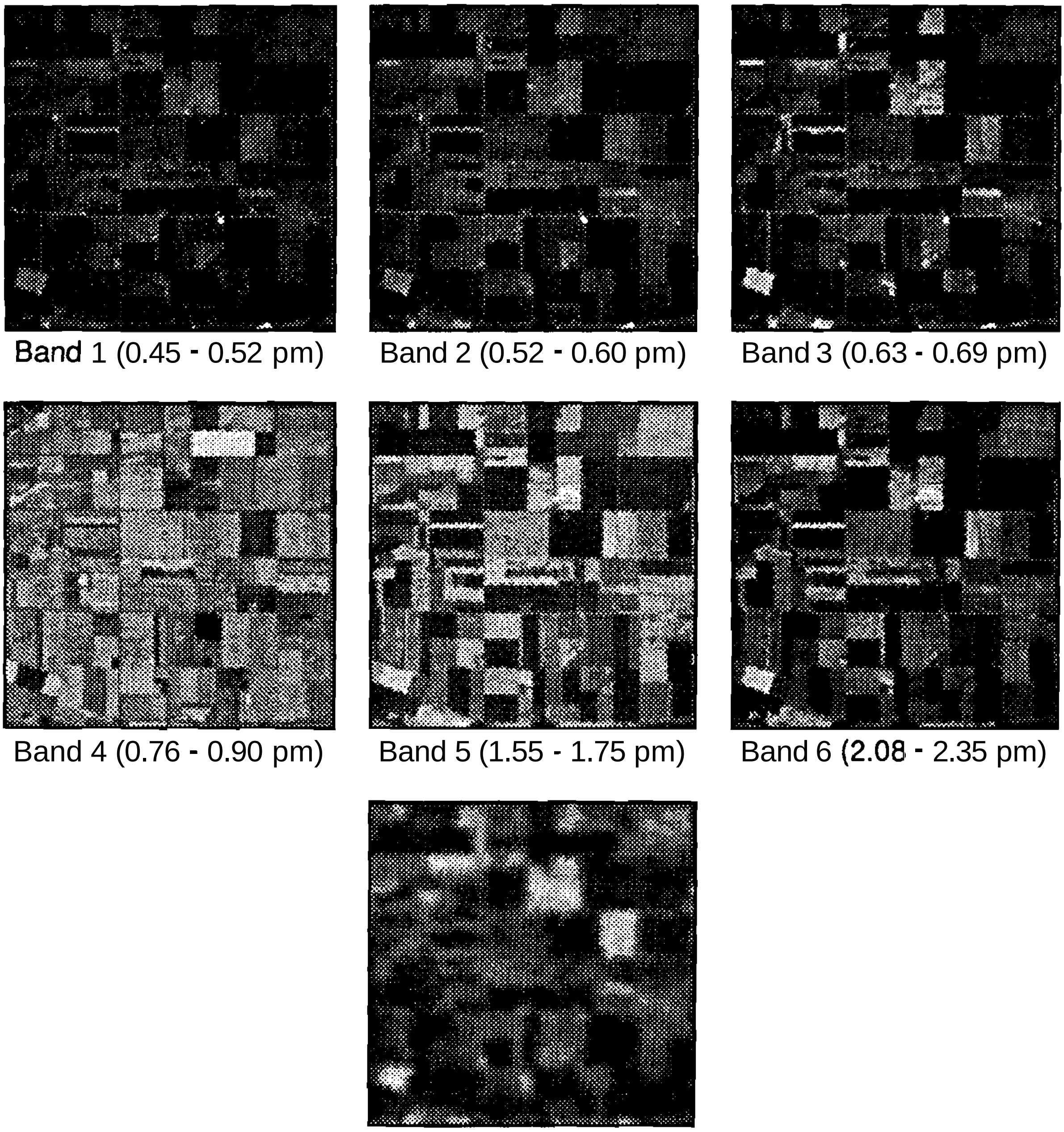
Figure 2.15 Corresponding Differences in the Total Classification Errors.

Figure 2.15 shows the corresponding differences in the total classification error without data reflection. The discrepancies in acceptance probabilities due to lack of data reflection in Fig. 2.14 cause as much as 5% difference in the total classification error in the region d < 1.5.

2.5.2 Experiment with Real Data

For a test with real data, a Landsat Thematic Mapper data set which was acquired over an agricultural area in Tippecanoe County, Indiana in July, 1986 was used with all seven features (*i.e.*, the dimensionality 9 = 7). From the available ground truth data, 4 different information classes - corn, soybeans, wheat and alfalfa/oats - were identified. About 10% of the samples were randomly selected from each information class to serve as training samples.

The total number of training samples were 2124, and there were 21,924 test samples. Figure 2.16 shows the July data set and Fig. 2.17 is the associated ground truth map.



Band 1 (0.45 - 0.52 pm)

Band 2 (0.52 - 0.60 pm)

Band 3 (0.63 - 0.69 pm)

Band 4 (0.76 - 0.90 pm)

Band 5 (1.55 - 1.75 pm)

Band 6 (2.08 - 2.35 pm)

Band 7 (10.4 - 12.5 pm)

Figure 2.16 July Thematic Mapper (TM) Data Set.

Since the information classes might consist of several sub-classes, clustering was performed on the training samples belonging to each information class to obtain a set of constituent sub-classes, each of which can be described with a multivariate normal distribution (Swain 78).

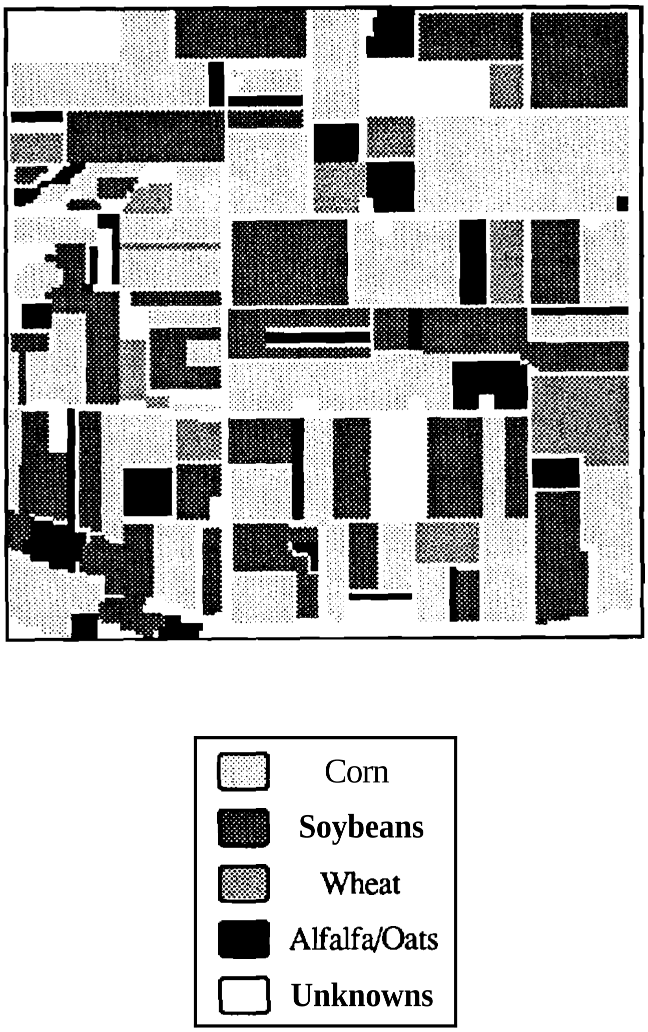


Figure 2.17 Associated Ground Truth Map.

Table 2.2 Training and Test Samples of Landsat Thematic Mapper Data.

Information Classes	Number of		
	Sub classes	Training Samples	Test Samples
Corn	2	913	9371
Soybeans	2	824	8455
Wheat	4	181	1923
Alfalfa/Oat	4	206	2175
Total	12	2124	21924

In a manner similar to the previous experiment with the simulated data, the acceptance probability was increased from **0.01** to **0.99** in steps of **0.01** to see how the omission and commission, class-averaged and total classification errors would change with respect to acceptance probabilities. The graphs of classification error versus acceptance probability are shown in Fig. **2.1**8 **- 2.20.**

The rate of decrease in omission error with respect to increasing acceptance probability can give some indication of how representative the training samples are. That is, if the training samples are very representative of the samples belonging to that class, then, the omission error will decrease almost ***linearly*** with respect to acceptance probability. The commission error curve also is able to show how separable the given class of interest is from the others class. Sub-class **2** of corn and sub-class **3** and **4** of wheat seemed to be much more separable than the others since the commission error curves were virtually not increasing with respect to increasing acceptance probability. Commission error increased rather sharply in all sub-classes of soybeans and alfalfa/oats.

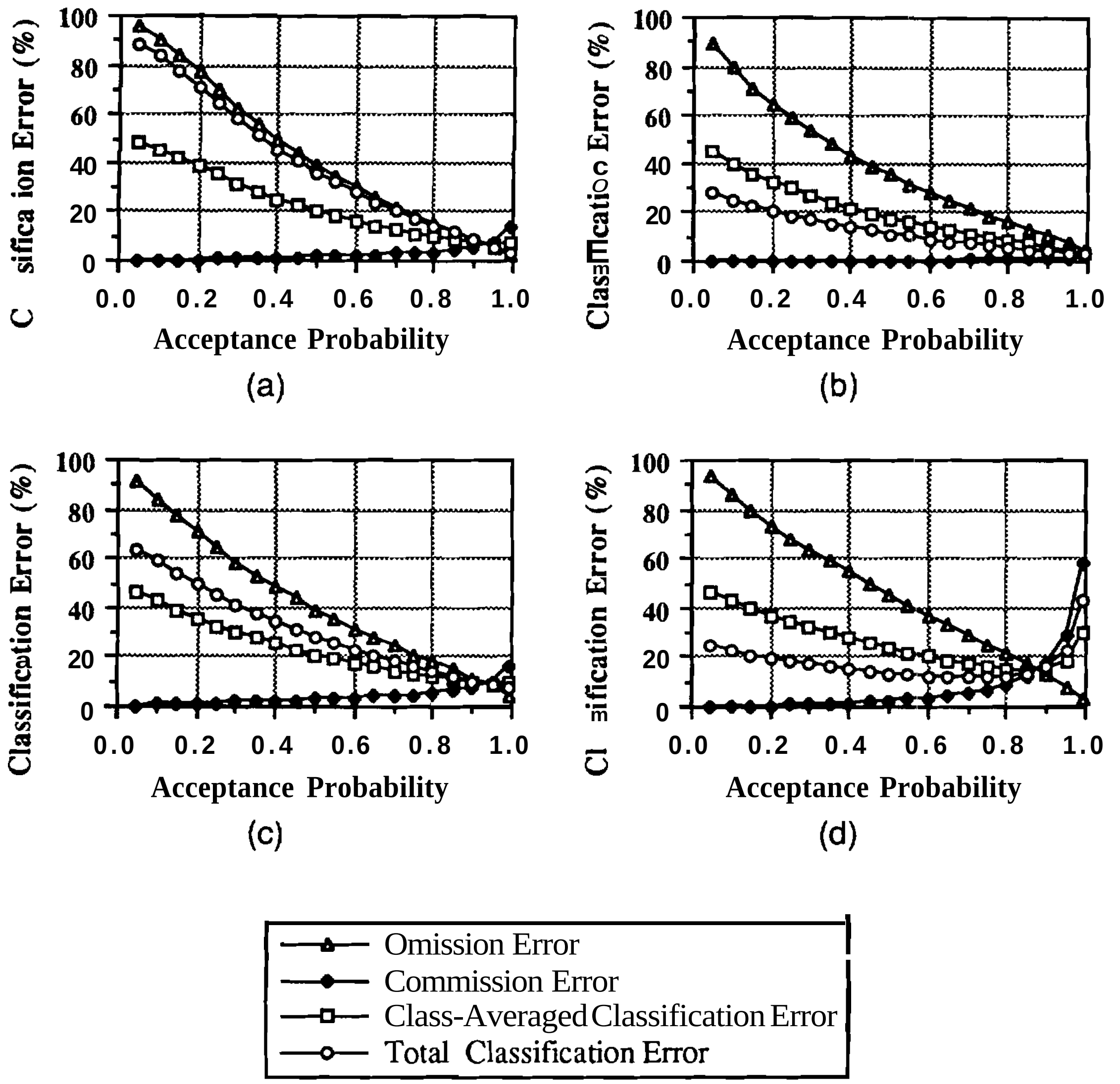


Figure 2.18 Classification Errors versus Acceptance Probability in Significance Testing With Landsat Thematic Mapper Data (Class; corn and soybeans). (a) Sub-class 1 of corn. (b) Sub-class 2 of corn. (c) Sub-class 1 of soybeans. (d) Sub-class 2 of soybeans.

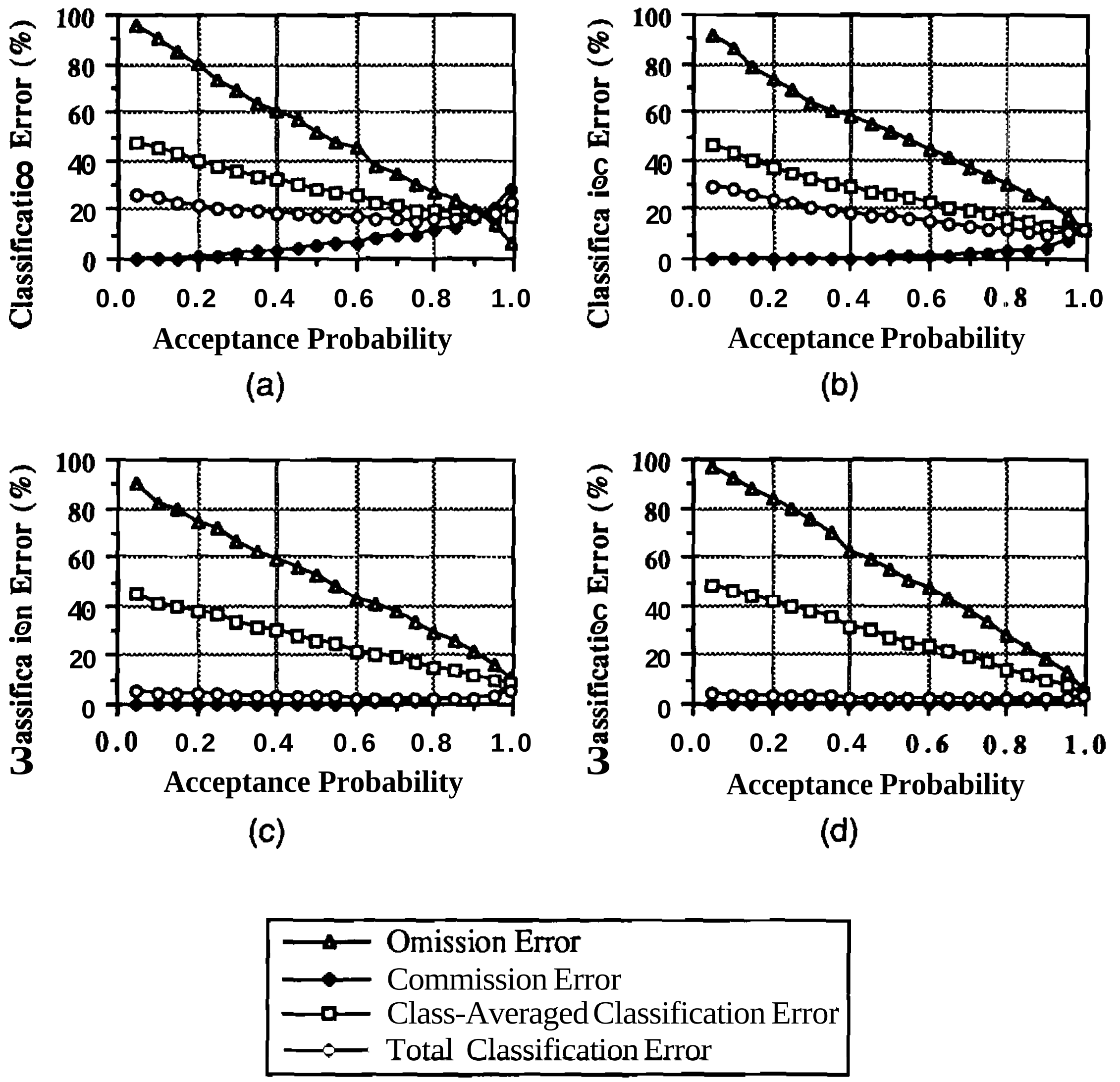
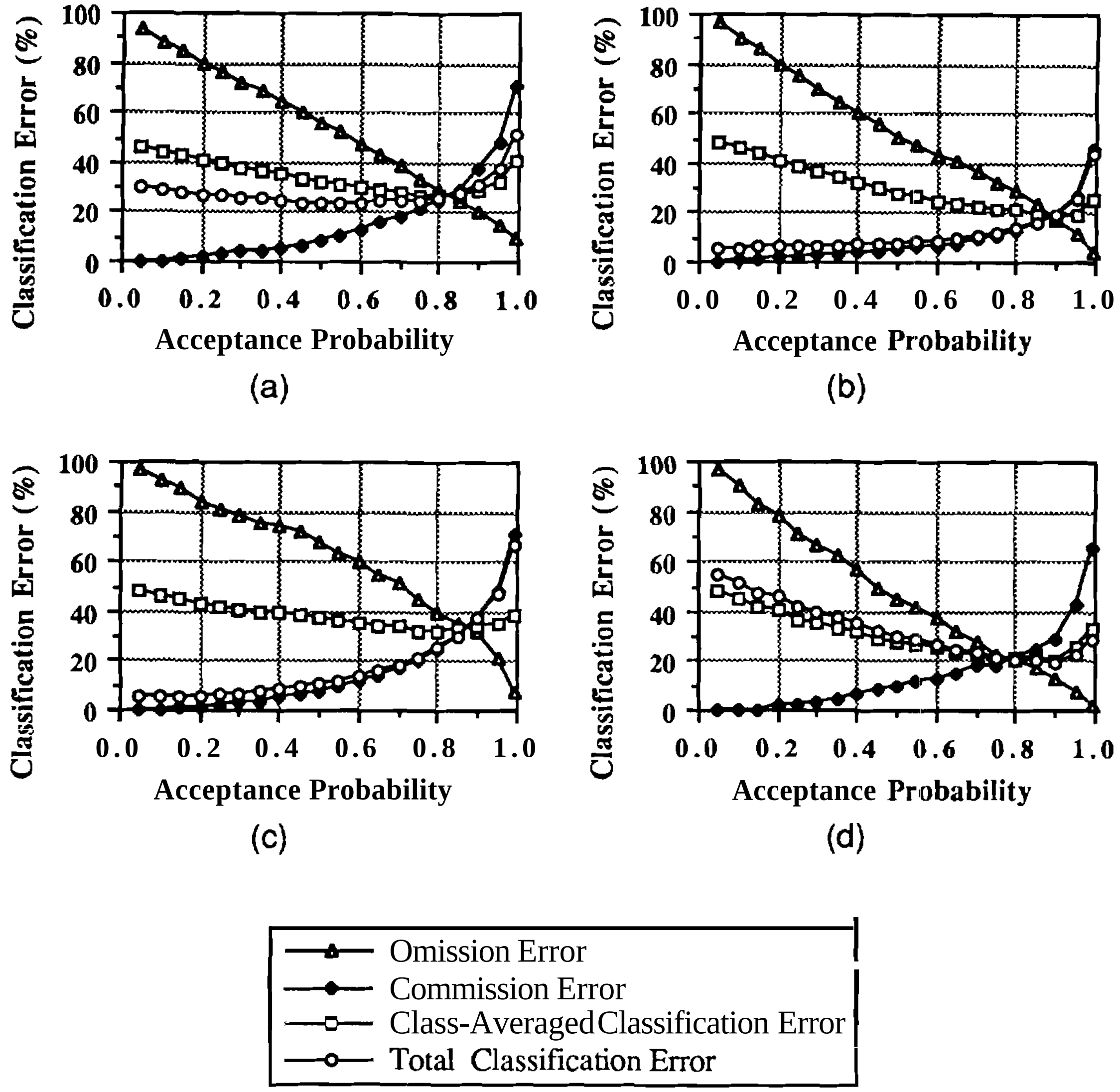


Figure 2.19 Classification Errors versus Acceptance Probability in Significance Testing with Landsat Thematic Mapper Data (Class wheat). (a) Sub-class 1 of wheat. (b) Sub-class 2 of wheat. (c) Sub-class 3 of wheat. (d) Sub-class 4 of wheat.



(a)

(b)

(c)

(d)

Figure 2.20 Classification Errors versus Acceptance Probability in Significance Testing with Landsat Thematic Mapper Data (Class alfalfa/oats). (a) Sub-class 1 of alfalfdoats. (b) Sub-class 2 of alfalfa/oats. (c) Sub-class 3 of alfalfdoats. (d) Sub-class 4 of alfalfa/oats.

The class-averaged classification error criterion in eq. (2.11.a) was applied to each information class to estimate the optimum acceptance probabilities at the sub-class level, and the results are shown in Table 2.3. The performances in information class level were obtained by assessing omission and commission errors after optimal acceptance probabilities were selected separately for each sub-class. A Gaussian kernel function was used in density estimation with the positive reflection technique. Although an optimal window size can be computed as in (Silverman 86), various window sizes (h=0.1 - h=0.6 in steps of 0.1) were examined to observe virtually no differences. The suggested optimal window sizes (Silverman 86) were in the range of 0.14 ~ 0.27. The results reported here were obtained with h=0.5.

In order to make a comparison with the estimated optimal acceptance probability, a specific value which attained the given optimality condition was manually determined as before. This value is referred as "optimum acceptance probability determined by scanning" in Table 2.3. As seen in the table, the estimated values of optimal acceptance probabilities using the proposed method agreed quite well with those manually selected. The class-averaged classification errors evaluated for each sub-class with estimated optimum acceptance probability were also very close to those obtained with the manually selected acceptance probabilities. The maximum difference between the estimated and the manually selected acceptance probabilities was only 0.03, except for the sub-class 4 of "alfalfa/oats" which had a difference of 0.08. The corresponding difference in the class-averaged classification error in this sub-class was only 0.34%.

Inspecting Fig. 2.20.(d) reveals that the class-averaged classification error was not changing much in the region of 0.7 < a c 0.9. The resulting class-averaged classification errors of the sub-classes with the estimated acceptance probabilities were all equal or slightly larger than those with manually selected optimal acceptance probabilities.

The proposed algorithm was also applied at the information class level as reported in Table 2.4, and its results were seen to be also very satisfactory since the acceptance probabilities deviated no more than 0.04 and the corresponding maximum difference in the class-averaged classification error was less than 1%.

Table *2.3* **Significance Testing of Landsat Thematic Mapper Data with the Class-Averaged Classification Error Criterion Applied at the Sub-class Level.**

All errors are in percent units

	Optimum acceptance probability determined by scanning					Estimated optimum acceptance probability				
Classes	α^*	$\varepsilon_1(\alpha^*)$	$\varepsilon_2(\alpha^*)$	$E_1(\alpha^*)$	$E_2(\alpha^*)$	$a*$	$\varepsilon_1(\alpha^*)$	$\varepsilon_2(\alpha^*)$	$E_1(\alpha^*)$	$E_2(\alpha^*)$
Corn 1	0.96	5.12	7.84	6.48	5.33	0.98	3.55	10.34	6.95	4.09
Corn 2	0.99	4.33	2.66	3.50	3.19	0.98	5.46	1.86	3.66	2.99
Corn	-	4.64	2.79	3.72	3.58	-	4.71	2.07	3.39	3.19
Soybeans 1	0.94	8.73	8.87	8.80	8.77	0.96	7.29	10.50	8.89	8.27
Soybeans 2	0.85	17.31	11.54	14.43	13.05	0.87	15.72	13.25	14.49	13.90
Soybeans	-	12.94	11.17	12.05	11.85	-	11.43	12.86	12.14	12.31
Wheat 1	0.97	10.57	22.53	16.55	19.35	0.95	14.16	20.08	17.12	18.50
Wheat 2	0.97	14.77	8.27	11.52	10.36	0.97	14.77	8.27	11.52	10.36
Wheat 3	0.99	10.31	7.24	8.77	7.49	0.96	15.32	4.36	9.84	5.25
Wheat4	0.99	6.39	2.97	4.68	3.10	0.99	6.39	2.97'	4.68	3.10
Wheat	-	10.45	5.38	7.92	5.82	-	12.27	4.65	8.46	5.31
Alfalfa/Oats 1	0.80	28.50	25.06	26.78	26.17	0.83	26.42	27.54	26.98	27.18
Alfalfa/Oats 2	0.90	17.28	19.63	18.45	19.50	0.89	18.66	18.73	18.70	18.73
Alfalfa/Oats 3	0.80	40.07	24.93	32.50	25.74	0.81	39.72	25.98	32.85	26.72
Alfalfa/Oats 4	0.79	20.93	20.36	20.65	20.69	0.88	14.92	27.06	20.99	20.13
Alfalfa/Oats	-	24.14	21.31	22.72	21.59	-	22.62	21.26	21.94	21.39

a : Optimum acceptance probability

$\varepsilon_1(\alpha^*)$: Omission error with the acceptance probability $a*$

$\varepsilon_2(\alpha^*)$: Commission error with the acceptance probability $a*$

$E_1(a')$: Class-averaged classification error with the acceptance probability a'

$E_2(\alpha^*)$: Total classification error with the acceptance probability a'

Table 2.4 **Significance Testing of Landsat Thematic Mapper Data with the Class-Averaged Classification Error Criterion Applied at the Information Class Level.**

All errors are in percent units

	determined by scanning					probability				
Classes	a'	$\varepsilon_1(a^*)$	$\varepsilon_2(\alpha^*)$	$E_1(a')$	$E_2(\alpha^*)$	a'	$\varepsilon_1(a')$	$\varepsilon_2(\alpha^*)$	$E_1(a')$	$E_2(\alpha^*)$
Corn 1	0.99	2.35	13.48	7.92	3.24	0.99	2.35	13.48	7.92	3.24
Corn 2	0.98	5.46	1.86	3.66	2.99	0.97	6.38	1.54	3.96	3.06
Corn	-	4.24	2.15	3.19	3.04	-	4.79	1.84	3.32	3.10
Soybeans 1	0.99	3.95	16.17	10.06	7.69	0.98	5.18	13.02	9.10	7.58
Soybeans 2	0.78	22.76	7.79	15.28	11.71	0.78	22.76	7.79	15.28	11.71
Soybeans	-	13.18	8.96	11.07	10.58	-	13.80	8.52	11.16	10.55
Wheat 1	0.99	6.34	28.05	17.19	22.27	0.97	10.57	22.53	16.55	19.35
Wheat 2	0.99	11.36	11.95	11.66	11.76	0.99	11.36	11.95	11.66	11.76
Wheat 3	0.99	10.31	7.24	8.77	7.49	0.95	16.16	3.56	9.86	4.59
Wheat4	0.99	6.39	2.97	4.68	3.10	0.96	12.43	1.62	7.03	2.05
Wheat	-	8.48	5.94	7.21	6.16	-	12.38	3.93	8.15	4.67
Alfalfa/Oats 1	0.98	11.05	61.21	36.13	44.97	0.95	15.37	48.14	31.76	37.53
Alfalfa/Oats 2	0.77	30.90	11.76	21.33	12.84	0.75	32.66	10.95	21.81	12.18
Alfalfa/Oats 3	0.51	66.55	7.94	37.24	11.10	0.54	64.81	9.07	36.94	12.08
Alfalfa/Oats 4	0.99	1.74	65.21	33.48	28.98	0.99	1.74	65.21	33.48	28.98
Alfalfa/Oats	-	23.40	14.84	19.12	15.68	-	24.97	13.80	19.38	14.90

a^* : Optimum acceptance probability

$\varepsilon_1(a')$: Omission error with the acceptance probability a'

$\varepsilon_2(\alpha^*)$: Commission error with the acceptance probability a'

$E_1(a')$: Class-averaged classification error with the acceptance probability a'

$E_2(\alpha^*)$: Total classification error with the acceptance probability $a*$

The class-averaged classification errors evaluated for each information class are compared in Fig. 2.21. Note that, as discussed in previous section, imposing the class-averaged classification error optimality criterion at the sub-class level didn't necessarily hold the same optimality at the information class level as seen in Fig. 2.21 in the corn and alfalfa/oats classes.

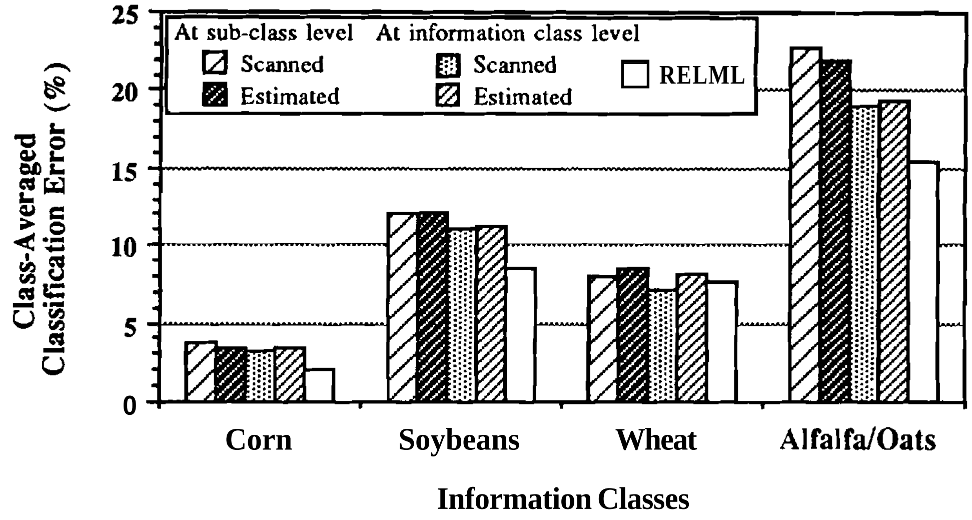


Figure 2.21 Comparisons of Class-Averaged Classification Errors Evaluated for Each Information Class.; Optimal acceptance probabilities were selected using the class-averaged classification error criterion. The first two columns for each information class show the class-averaged classification errors based on the optimal acceptance probabilities at the sub-class level. The next two columns correspond to the case when the optimal acceptance probabilities are acquired at the information class level. "REL-ML" is the result obtained with a (relative) maximum likelihood classifier designed with all 12 sub-classes.

With all 12 sub-classes and their class statistics, a relative maximum likelihood classifier in the original seven dimensional space was designed and its result (denoted by "REL-ML") is also included in Fig. 2.21 to show the effect of dimensionality reduction. In the corn and wheat classes, there seemed to be not much information loss due to dimensionality reduction. However, there was as much as 3 - 5% of class-averaged classification error increase in soybeans and alfalfa/oats.

Finally, the total classification error criterion was used with the relative weight A = 1 and these results are presented in Table 2.5.

Table *2.5* **Significance Testing of Landsat Thematic Mapper Data with the Total Classification Error Criterion.**

All errors are in percent units

	Optimum acceptance probability determined by scanning					Estimated optimum acceptance probability				
Classes	α^*	$\varepsilon_1(a')$	$\varepsilon_2(\alpha^*)$	$E_1(a')$	$E_2(\alpha^*)$	a'	$\varepsilon_1(a')$	$\varepsilon_2(\alpha^*)$	$E_1(a')$	$E_2(\alpha^*)$
Corn 1	0.99	2.35	13.48	7.92	3.24	0.99	2.35	13.48	7.92	3.24
Corn 2	0.98	5.46	1.86	3.66	2.99	0.97	6.38	1.54	3.96	3.06
Corn	-	4.24	2.15	3.19	3.04	-	4.79	1.84	3.32	3.10
Soybeans 1	0.98	5.18	13.02	9.10	7.58	0.96	7.29	10.50	8.89	8.27
	0.74	25.63	6.59	16.11	11.57	0.67	31.40	4.84	18.12	11.78
Soybeans	-	15.21	7.49	11.35	10.45	-	19.11	5.63	12.37	10.79
Wheat 1	0.76	29.18	10.11	19.65	15.19	0.76	29.18	10.11	19.65	15.19
Wheat 2	0.91	21.21	4.67	12.94	9.99	0.91	21.21	4.67	12.94	9.99
Wheat3	0.80	28.97	1.14	15.06	3.43	0.74	33.43	0.80	17.11	3.47
Wheat4	0.86	20.96	0.78	10.87	1.58	0.69	39.08	0.35	19.71	1.88
Wheat	-	24.54	1.67	13.11	3.67	-	30.68	1.31	16.00	3.87
Alfalfa/Oats 1	0.58	48.88	11.25	30.06	23.43	0.48	57.34	7.61	32.47	23.71
Alfalfa/Oats 2	0.00	100.00	0.00	50.00	5.64	0.00	100.00	0.00	50.00	5.64
Alfalfa/Oats 3	0.07	94.77	0.12	47.45	5.23	0.03	97.91	0.08	48.99	5.36
Alfalfa/Oats 4	0.89	13.57	28.09	20.83	19.80	0.89	13.57	28.09	20.83	19.80
Alfalfa/Oats	-	65.20	1.26	33.23	7.56	-	67.86	1.03	34.45	7.62

a^*	: Optimum acceptance probability
$\varepsilon_1(\alpha^*)$	: Omission error with the acceptance probability a'
$\varepsilon_2(\alpha^*)$	: Commission error with the acceptance probability a'
$E_1(a^*)$	: Class-averaged classification error with the acceptance probability a'
$E_2(\alpha^*)$	: Total classification error with the acceptance probability a'

Notice that applying this error criterion at the sub-class level always attains the same optimality at the information class level, too. In most of the sub-classes, as in previous cases, the estimated optimal acceptance probabilities were very close to those manually selected. In the case of sub-classes 2, and 3 of alfalfa/oats, optimal acceptance probabilities were found to be very small since the total classification errors in these sub-classes were rapidly increasing with respect to acceptance probabilities.

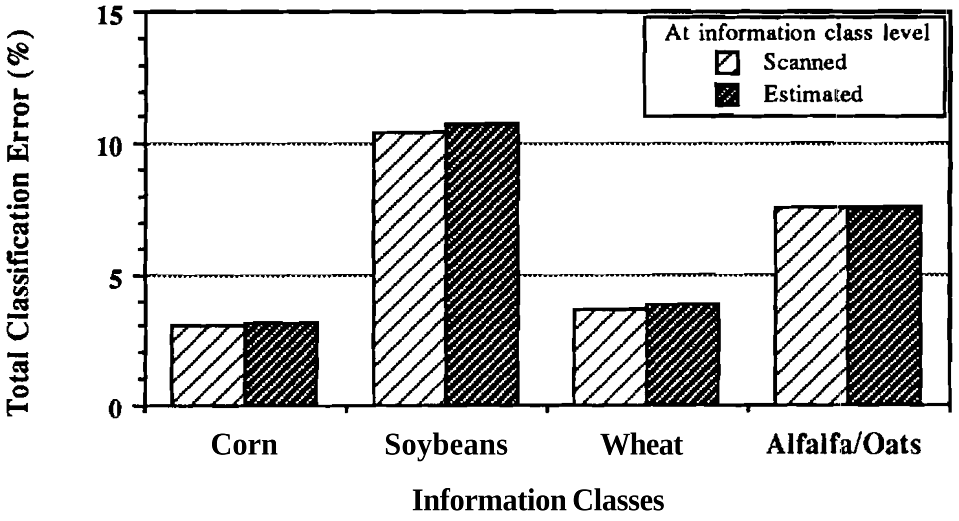


Figure 2.22 **Comparison of Total Classification Errors Evaluated at the Information Class Level.; Optimal acceptance probabilities were selected using the total error criterion.**

Total classification errors evaluated for each information class are presented in Fig. 2.22 which shows a very good matches between total classification errors obtained with "true" and estimated optimal acceptance probabilities.

For visual comparison of performances, Fig. 2.23 ~ 2.26 show the locations of the samples identified by the significance testing and the relative maximum likelihood classifier (REL-ML) which is included to see the effect of dimensionality reduction of feature vectors.

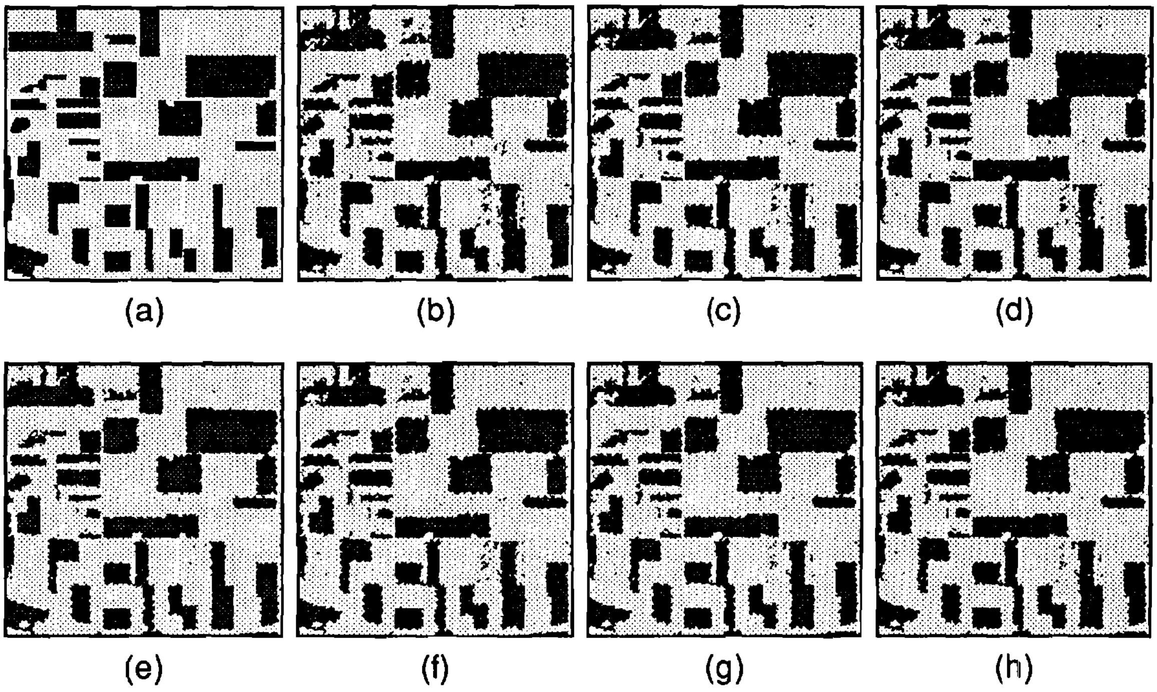


Figure 2.23 Results for the Class "Corn" Samples. (a) Ground truth location of corn samples. (b) Significance testing with acceptance probability manually selected under the class-averaged error criterion at sub-class level. (c) Significance testing with acceptance probability manually selected under the class-averaged error criterion at information class level. (d) Significance testing with acceptance probability manually selected under the total error criterion. (e) Relative maximum likelihood classifier (REL-ML). (f) Significance testing with acceptance probability estimated under the class-averaged error criterion at sub-class level. (g) Significance testing with acceptance probability estimated under the class-averaged error criterion at information class level. (h) Significance testing with acceptance probability estimated under the total error criterion.

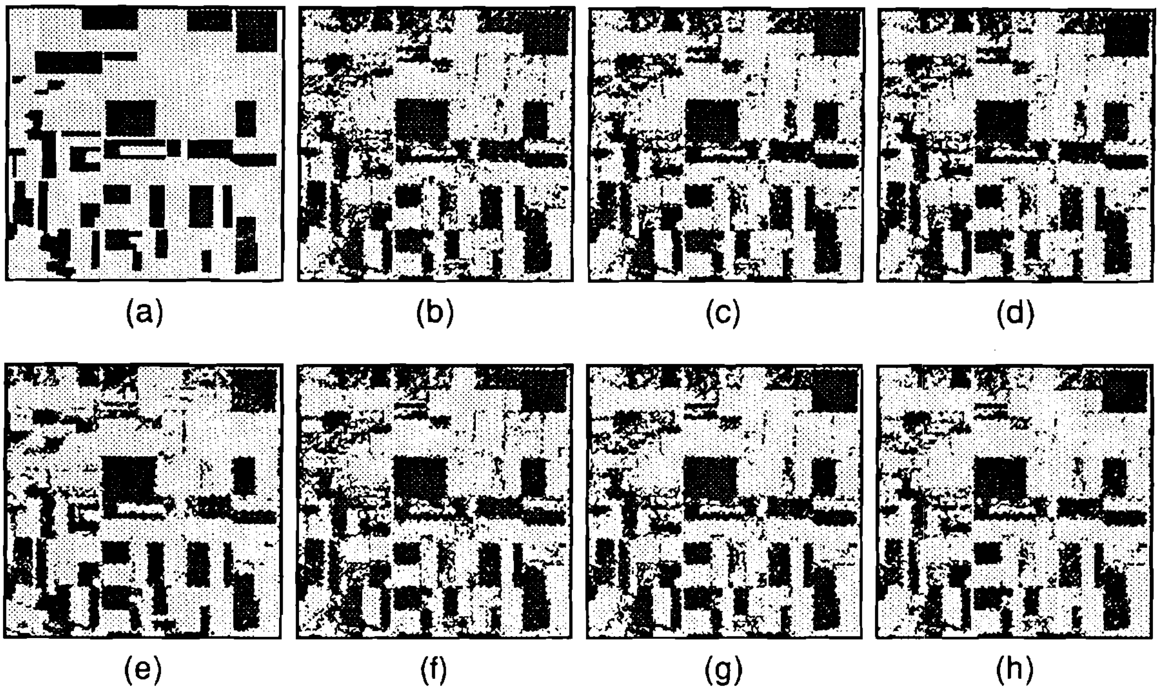


Figure **2.24** Results for the Class "Soybeans" Samples. (a) Ground truth location of soybeans samples. (b) Significance testing with acceptance probability manually selected under the class-averaged error criterion at sub-class level. (c) Significance testing with acceptance probability manually selected under the class-averaged error criterion at information class level. (d) Significance testing with acceptance probability manually selected under the total error criterion. (e) Relative maximum likelihood classifier (REL-ML). (f) Significance testing with acceptance probability estimated under the class-averaged error criterion at sub-class level. (g) Significance testing with acceptance probability estimated under the class-averaged error criterion at information class level. (h) Significance testing with acceptance probability estimated under the total error criterion.

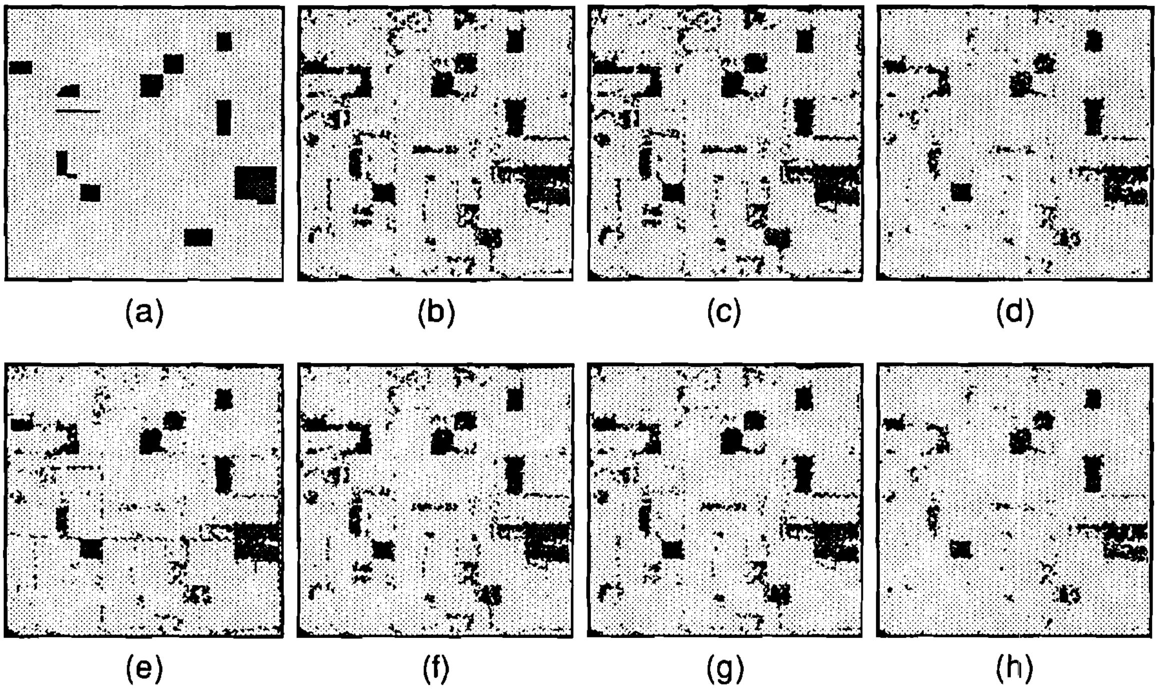


Figure 2.25 Results for the Class "Wheat" Samples. (a) Ground truth location of wheat samples. (b) Significance testing with acceptance probability manually selected under the class-averaged error criterion at sub-class level. (c) Significance testing with acceptance probability manually selected under the class-averaged error criterion at information class level. **(d)** Significance testing with acceptance probability manually selected under the total error criterion. (e) Relative maximum likelihood classifier (REL-ML). (f) Significance testing with acceptance probability estimated under the class-averaged error criterion at sub-class level. (g) Significance testing with acceptance probability estimated under the class-averaged error criterion at information class level. (h) Significance testing with acceptance probability estimated under the total error criterion.

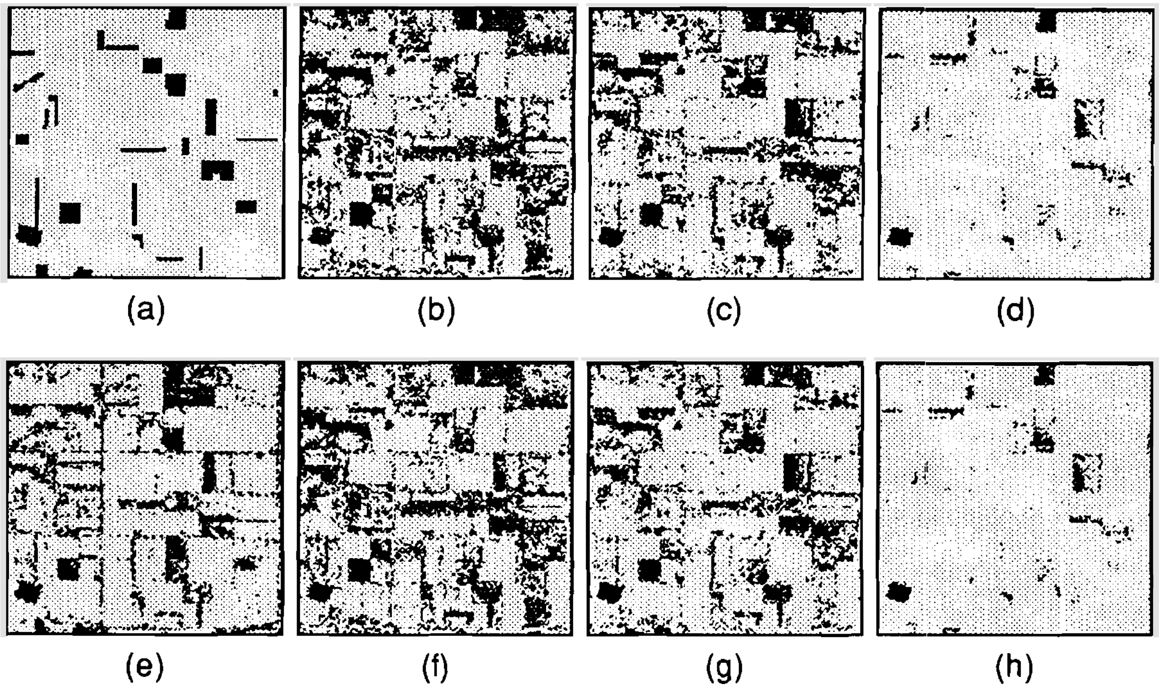


Figure 2.26 Results for the Class "Alfalfa/Oats" Samples. (a) Ground truth location of alfalfa/oats samples. (b) Significance testing with acceptance probability manually selected under the class-averaged error criterion at sub-class level. (c) Significance testing with acceptance probability manually selected under the class-averaged error criterion at information class level. (d) Significance testing with acceptance probability manually selected under the total error criterion. (e) Relative maximum likelihood classifier (REL-ML). (f) Significance testing with acceptance probability estimated under the class-averaged error criterion at sub-class level. (g) Significance testing with acceptance probability estimated under the class-averaged error criterion at information class level. (h) Significance testing with acceptance probability estimated under the total error criterion.

The significance testing procedure produced reasonably good classification maps, except for the classes, soybeans and alfalfaloats, compared to the relative maximum likelihood classifier, which not only requires a complete list of classes to be defined and their class statistics computed, but also classifies samples in the original feature space without dimensionality reduction. The estimated optimum acceptance probabilities produced classification maps which were hardly differentiable with those obtained with manually selected optimum acceptance probabilities.

To see the effect of data reflection on the estimation of acceptance probabilities, the same experiments with both class-averaged and total classification error criteria were performed with varying window sizes (h= 0.1 ~ 0.6 in steps of 0.1) without data reflection. There were observed no differences except of sub-class 3 of alfalfa/oats with h=0.6 where the optimum accept probability without data reflection was estimated as 0 instead of 0.03 under the total classification error criterion. Data reflection would change the density estimate values only where the $(x^Tx)^{0.5}$ values are less than about 3-4 times the selected window size, h, due to the exponential term in the Gaussian kernel function. Considering a $(x^Tx)^{0.5}$ value corresponding to an acceptance probability a= 0.5 in the seven dimensional space is 2.52, which is only comparable to 4 times the largest window size h=0.6. In most of the sub-classes except for sub-classes 2 and 3 of alfalfaloats, the estimated acceptance probabilities would not be affected by data reflection since the estimated acceptance probabilities were mostly much larger than 0.5.

2.6 Conclusions

In this chapter, a problem of estimating the optimal acceptance probability in significance testing was addressed. Estimating the optimal acceptance probability using a given data set should be very useful in applying a significance testing procedure. As optimality criteria, both class-averaged classification error and generalized total classification error criteria were considered. It is shown that the generalized total classification error criterion applied to each sub-class also attains the same optimality at the information class level. To have an optimal class-averaged classification error criterion at the information class level, however, the generalized total classification error criterion with a relative weight should be applied to each sub-class. If the class of interest doesn't need to be decomposed into sub-classes, the class-averaged classification error criterion can be applied even without the knowledge of prior probabilities. A data reflection technique required in mixture density estimation was observed to be useful when the underestimation of a density function in the region near 0 in the one-dimensional space of the selected test statistic causes overestimation of optimal acceptance probabilities. This estimation algorithm for acceptance probability should be very useful when one doesn't have enough prior knowledge about the data set to select the proper acceptance probability. This automatic estimation procedure can replace the lengthy and tedious process of manual selection of acceptance probability especially when the given class of interest consists of a large number of sub-classes.

CHAPTER 3

PARTIALLY SUPERVISED CLASSIFICATION WITH UNSUPERVISED CLUSTERING

3.1 Introduction

In this chapter, partially supervised classification with only one known class is formulated as a relative classification problem. Advantages of both reduced requirements for necessary prior knowledge in an absolute scheme and the potentially robust and powerful discriminating capability of a relative one are sought by developing an automatic mechanism for extracting statistical information corresponding to the "others" class without recourse to prior knowledge supplied by 'the data-analyst. Even though the classifiers to be proposed in this chapter make decisions ultimately on a relative basis, the terminology "absolute" classifier will be interchangeably used with partially supervised classifier to emphasize its reduced dependence on prior knowledge.

The class "others" are decomposed into a set of sub-classes so that the density function of each sub-class can be modeled with a known parametric density function, for example, with the Gaussian density function. This decomposition is achieved through a weighted unsupervised clustering procedure which subsequently develops the unknown class definitions and their corresponding class statistics through a unsupervised clustering. Once the class statistics of the constituent components of the "others" class are found, conventional relative classifier such as a maximum likelihood classifier can be usecl to identify the samples belonging to the class of interest.

3.2 Partially Supervised Classification with Unsupervised Clustering

The given partially supervised classification problem with only one known class, which is the only class of interest, can be considered as an L class relative classification problem with unknown number L. The unknown sub-classes pertaining to the "others" and their statistical characteristics are developed using unsupervised clustering. Once the class statistics are developed, any relative classification scheme can be put to use. This problem is different from that of general unsupervised clustering in the following senses: (1) One is interested in finding samples of only one particular cluster (or class) and, one has prior statistical information, such as the probability density function of that class, or has a representative set of training samples of that class from which the statistical properties can be estimated. (2) The clusters corresponding to the class "others" do not need to be meaningful as useful informational classes and, the confusion between those clusters are not important as long as they are differentiable from the class of interest. Under this approach, the mixture density $f_X(x)$ is written as a weighted sum of L probability density functions as,

$$f_X(x) = \sum_{k=1}^{L} \pi_k \, f_X(x \mid C_k) \qquad (3.1)$$

$$\text{where,} \quad \pi_1 + \text{---} + \pi_L = 1$$

and π_k and $f_X(\bullet \mid C_k)$ are the prior probability and probability density function of the $k^{\underline{th}}$ class, respectively, k = 1, --- , L. The notation of C_1 and C_2, ---, C_L means that $C_1 = C_{int}$ and C_2, ---, C_L are the sub-classes of C_{others} which will be found through unsupervised clustering. In the given partially supervised classification, only $f_X(x|C_1)$ is known.

Any unsupervised clustering procedure (Fukunaga 90) can be used to decide the number of classes, L and to obtain the initial specification of clusters which can initiate subsequent supervised clustering. Special care should be taken so that there is no confusion between C_{int} and the clusters corresponding to C_{others}. In other words, the cluster statistics of C_{others} should not be biased by the samples belonging to C_1. One conceivable approach for reducing the bias is to find the clusters of C_{others} by performing clustering with a subset of data in which a significant portion of the C_1 samples are removed through significance testing. In addition to the difficulty in selecting the proper significance level, however, the approach still has the bias problem, especially when C_{others} is not well separated from C_1. Instead of removing the effect of C_1 samples in a rather absolute way, it is possible to assign to each sample a weight factor which is related to the relative likelihood of belonging to C_{others} and to use it in the unsupervised clustering.

Let the weight denoted by $\overline{w}_{i1}$ in eq. (3.2.a) indicate the relative likelihood of sample x_i being to C_{others}.

$$\overline{w}_{i1} \equiv 1 - w_{i1} \qquad (3.2.a)$$

$$\text{where,} \quad w_{i1} = \pi_1 \frac{f_X(x_i \mid C_1)}{f_X(x_i)} \qquad (3.2.b)$$

Note that evaluating the weight factor, $\overline{w}_{i1}$ requires 'the additional knowledge of π_1 (or N_1 since $\pi_1 = N_1 / N$, where N_1 is the total number of samples belonging to the class of interest.) and the mixture density $f_X(x_i)$'s. Assume for now that the prior π_1 (or, N_1) is available (the estimation of N_1 will be discussed later). Since the purpose of this unsupervised clustering is to provide an initial specification of clusters to initiate the clustering process, an exact evaluation of the probability density ratio in eq. (3.2.b) would not be necessary. A direct estimation of $f_X(x_i)$ through non-parametric density estimation, would require complex computation, but an approximation can be obtained by noting that w_{i1} can be expressed as a ratio,

$$w_{i1} = \frac{N_1\, f_X(x_i \mid C_1)\Delta V}{N\, f_X(x_i)\Delta V} \qquad (3.3)$$

Assume a data point x_i is inside a hypersphere with volume AV. Then, $Nf_X(x_i)\Delta V$ in eq. (3.3) can be approximated by,

$$Nf_X(x_i)\Delta V \approx N\int_{x\in\Delta V} f_X(x)dx \qquad (3.4)$$

The right side of eq. (3.4) is the expected number of data samples found inside the hypersphere. Therefore, the approximate value for it can be obtained by counting the number of samples in the hypersphere. In the same way, the numerator in eq. (3.3) can be approximated by,

$$N_1\ f_X(x_i\ |\ C_1)\Delta V \approx N_1\int_{x\in\Delta V} f_X(x|C_1)dx \qquad (3.5.a)$$

This is the expected number of samples from the class C_1, found inside the volume AV. This can be computed using the known probability density function $f_X(x|C_1)$. Instead of discretizing the whole feature space by picking a certain value of AV and counting the data points inside the hyperspheres, a simple clustering routine using a Euclidean distance measure is used to find a set of hyperspheres which can *effectively* cover the feature space, as in Fig. 3.1.

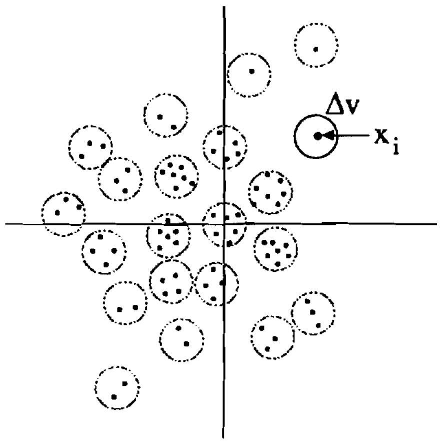


Figure 3.1 **Computation of Weights Using Clustering.; Clustering is performed to find a set of hyperspheres *effectively* covering the feature space.**

With an appropriate clustering condition, unsupervised clustering can be performed to divide the feature space into a set of small hyperspheres which cover effectively all given data samples. The critical distance for creating a new cluster is set up in such a way that each hypersphere corresponds to a cluster, and inside the hyperspheres, the probability density functions, $f_X(x)$ and $f_X(x|C_1)$ should not change much. In each hypersphere or cluster, eq. (3.5.a) is computed as,

$$N_1\ f_X(x_i\ |\ C_1)\Delta V \approx N_1\ f_X(x_0|C_1)\Delta V \qquad (3.5.b)$$
$$\text{where, } x_0 \in \Delta V$$

x_0 is a location inside the given hypersphere. The cluster mean is used for x_0 when the probability density $f_x(\bullet|C_1)$ is evaluated. While the value in eq. (3.5.b) is an approximation of the expected number of C_1 data points inside the given hypersphere, the count of data points inside the hypersphere is an approximated value from eq. (3.4). The weight factor is computed using these two values in eq. (3.2.a) and eq. (3.3) and this same weight value is assigned to all the data points inside that hypersphere.

With these weight factors, an unsupervised clustering is performed to find the initial clusters corresponding to C_{others}. Since the weight $\overline{w}_{i1}$ in eq. (3.2.a) indicates the relative likelihood that a data sample is from C_{others}, data samples of C_1 will have very small weights. Any cluster which has most samples with negligible weight factors should be deleted since the samples in it are mainly from C_1. Therefore, the unsupervised clustering with these weights can avoid the potential influence of the data points belonging to C_1 upon new clusters of C_{others}. For each cluster k corresponding to C_{others}, (that is, k = 2, ---, L), the ***effective*** number of elements in the cluster, N_k is computed as a sum of the weights of data samples in the $k^{\underline{th}}$ cluster as,

$$N_k = \sum_{i \in I_k} \overline{w}_{i1} \tag{3.6.a}$$

where I_k is the index set of the $k^{\underline{th}}$ cluster (i.e., if $i \in I_k$, then $x_i \in C_k$). This ***effective*** number will indicate the possibility of being part of C_{others}. Any cluster with a negligible effective number of members is deleted.

$$R_k = \frac{N_k}{\text{Number of samples in cluster } C_k} \tag{3.6.b}$$

The ratio of the effective number to the actual sample number assigned to the cluster in eq. (3.6.b) is also checked, and any cluster with a small value of this ratio is deleted since most samples In the cluster have negligible weight factors. When the number of class-of-interest samples, N_1 is under-estimated, this ratio checking is very important since there are extraneous clusters generated in the region where most of the class-of-interest samples are located. This ratio checking should also be effective when the class-of-interest samples are distributed slightly differently from the known distribution function in some hyperspheres so that the numbers computed with eq. (3.5.b) deviate from those statistically expected. Without the ratio-checking, smaller values of eq. (3.5.b) in some hyperspheres than they should be would allow generating clusters of C_{others} which would take a significant portion of class-of-interest samples.

The ***effective*** cluster mean M_k is computed as,

$$M_k = \frac{1}{N_k} \sum_{i \in I_k} \overline{w}_{i1}\, x_i \tag{3.6.c}$$

Note that the influence of data point x_i on the cluster means of C_{others} is accordingly weighted by $\overline{w}_{i1}$. If second order statistics are necessary for clustering, then, the ***effective*** cluster covariance can also be computed with weights in a similar fashion. A few iterations of this unsupervised clustering with weights will suffice to provide a list of clusters corresponding to C_{others} and their initial specifications for the subsequent supervised clustering process.

Once the number of clusters and the specifications of the clusters are obtained through unsupervised clustering with weights, then a supervised clustering procedure can be started to develop the unknown class statistics. The class statistics developed are used in the relative classification scheme chosen. In certain cases, especially in analyzing high dimensional feature vectors, second order statistics, which are usually characterized by interband correlation structures, provide very crucial information to use in classification or clustering. In this case, a conventional clustering procedure such as the ISODATA (Hall and Ball 65) algorithm is not likely to perform well in developing class statistics since the algorithm does not account for interband correlation in the data set. Thus, a clustering based on the EM algorithm (Titterington et *al.* 85, Redner and Walker 84, Dempster *et al.* 77) can be used. That is, in the $m^{\underline{th}}$ iteration of clustering, weight factors, $w_{ik}[\widehat{\psi}^{(m)}]$, for i = 1,---, N and k = 1, ---, L, are computed as, (Expectation - step),

$$w_{ik}[\widehat{\psi}^{(m)}] = \frac{\widehat{\pi}_k^{(m)}\, \hat{f}_x^{(m)}(x_i \mid C_k)}{\sum_{j=1}^{L} \widehat{\pi}_j^{(m)}\, \hat{f}_x^{(m)}(x_i \mid C_j)} \tag{3.7}$$

where $\hat{f}_x^{(m)}(x_i \mid C_1) = f_x(x_i \mid C_1)$ for all m, and ψ is the set of parameters of the unknown probability density functions. For example, if the unknown probability density functions are Gaussian, then $\psi \equiv [\pi_2, ---, \pi_L, M_2, ---, M_L, \Sigma_2, ---, \Sigma_L]$. With the weight in eq. (3.7), a new maximum likelihood estimate of ψ, (*i.e.*, $\hat{\psi}^{(m+1)}$) is obtained (Maximization - step). These two steps are iteratively performed until convergence. Each iteration of these two steps is known to increase the joint likelihood of data samples (Titterington *et al.* 85, Redner and Walker 84, Dempster *et al.* 77). After convergence, the estimates of ψ specify the probability density functions of the clusters which can be used in the subsequent relative classification.

In summary, a flowchart of the proposed classifier based on clustering is shown in Fig. 3.2.

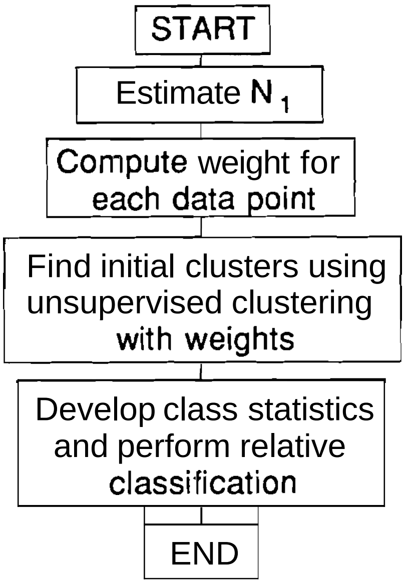

START
Estimate N_1
Compute weight for each data point
Find initial clusters using unsupervised clustering with weights
Develop class statistics and perform relative classification
END

Figure 3.2 **Flowchart of a Partially Supervised Classification with Unsupervised Clustering.**

3.3 Estimating the Number of Class-of-Interest Samples

In order to have the initial cluster definition in the previous section, it is required to know N_1, the number of samples belonging to the class of interest. Due to the limited prior knowledge and approximation involved in the estimation process, typically, an accurate estimation of this is another difficult task. Therefore, it will be very desirable to design a partially supervised classification algorithm which is not critically dependent on the estimate of this unknown. The objective of this section is to obtain a simple and reasonable estimate of N_1 which can produce a meaningful initial cluster definition, rather than developing a very rigorous estimation algorithm.

The unknown number N_1 will be estimated by "matching" two functions. Note that the probability density function $f_X(x|C_1)$ can be estimated in two different ways. It is typically estimated from the training samples supplied by user. But it can be also computed using the mixture density estimate $f_X(x)$ if the probability density function of C_{others}, $f_X(x|C_{others})$ and π_1 are available. Note that the mixture density $f_X(x)$ in eq. (3.1) can be written as,

$$f_X(x) = \pi_1 f_X(x|C_1) + (1-\pi_1) f_X(x|C_{others}) \qquad (3.8.a)$$

$$\text{where, } (1-\pi_1) f_X(x|C_{others}) = \sum_{k=2}^{L} \pi_k f_X(x \mid C_k)$$

In a specific region where the second term in eq. (3.8.a) is negligible compared to the first term, the estimate of $f_X(x|C_1)$ can be evaluated from the estimate $f_X(x)$ if a specific value is assumed for π_1. It will be a function of π_1. These two estimates can be matched together to find the best π_1. The function to be matched needs not be necessarily $f_X(x|C_1)$ even though it is a natural choice in the given partially supervised classification problem where prior information, other than the class statistics of C_1, is non-existent. Any function derivable from it can be matched.

To be general, denote the function to be matched as $h(x)$. This function is selected in such a way that it can be both evaluated from the probability density function of C_1 and estimated using the given data set if the prior probability of C_1 is available. Therefore, the estimate of $h(x)$ should be computable using the unlabeled samples when a certain value is assumed for the unknown prior probability. The estimate, based on a specific value of π_1, is denoted as $h(x|\pi_1)$. It is compared with the function $h(x)$ evaluated using $f_X(x|C_1)$, which is estimated from the training samples, to find the unknown prior probability which causes $h(x|\pi_1)$ to be nearest to $h(x)$. This matching doesn't necessarily need to take place in the original feature space. It can be accomplished in any space derived from the original space. The measure of closeness of $h(x)$ and $h(x|\pi_1)$ should be defined according to the specific function $h(x)$ chosen. In this paper, $f_X(x|C_1)$ is selected as $h(x)$. Over the sub-space in which the unknown quantity, $(1-\pi_1)f_X(x|C_{others})$ in eq. (3.8.a), is negligible compared to $\pi_1 f_X(x|C_1)$, the estimate $h(x|\pi_1)$ is approximated as,

$$h(x \mid \pi_1) \approx \frac{f_X(x)}{\pi_1} \qquad (3.8.b)$$

As for the measure of closeness, the expected squared error may be used, with a weight function $w(x)$ included to account for the possibility of different weights for different x's.

$$\mathrm{Error}(\pi_1) = E_X\left\{\left[h(x \mid \pi_1) - h(x)\right]^2 w(x)\right\} \qquad (3.8.c)$$

The expectation of the weighted squared error is taken over the entire feature space, or over the selected subspace as required. With the approximation of eq. (3.8.b), the expectation is computed only over the region where eq. (3.8.b) remains valid. Equation (3.8.c) can be equivalently written as a function of N_1 explicitly as follows.

$$\mathrm{Error}(N_1) = E_X\left\{\left[Nf_X(x) - N_1 f_X(x|C_1)\right]^2 w(x)\right\} \qquad (3.8.d)$$

This is a matching process of weighted probability density functions, $Nf_X(x)$ and $N_1 f_X(x|C_1)$. In the case of the multivariate Gaussian distribution of $f_X(x|C_1)$, it is possible to know the region where most of the samples are located. Note that this matching process can be also performed in the one dimensional y space where $y = x^T x$. If the dimensionality is not high, it is possible to select an appropriate y_0 and corresponding region specified by $x^T x \leq y_0$ where most of C_1 samples are found. Therefore, with a suitable value of y_0, the matching of eq. (3.8.d) can be processed.

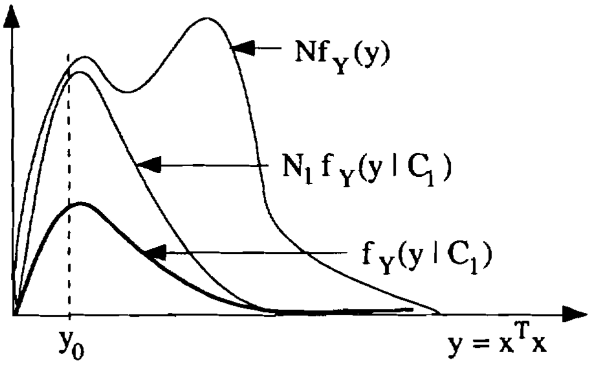


Figure 3.3 **Matching of Two Weighted Probability Density Functions.; $Nf_Y(y)$ is matched with $N_1 f_Y(y|C_1)$ over the region $0 \le y \le y_0$, $y = x^T x$ to find the best N_1. $f_Y(y)$ is the probability density function of y's. $f_Y(y|C_1)$ is the density function in y space corresponding to C_1 samples. y_0 is a user-specified threshold.**

An illustration in the y space is shown in Fig. 3.3 where the density function $f_X(x|C_1)$ is assumed to be the standard multivariate Gaussian.

The matching is performed in the region $0 \le y \le y_0$ where y_0 is a user-specified value indicating the region where the approximation of eq. (3.8.b) holds.

The unknown number N_1 is determined to minimize the expected error between $h(x)$ and $h(x|\pi_1)$ as,

$$\hat{N}_1 = N \frac{\int \left[w(x) f_X(x) f_X(x|C_1)\right] f_X(x) dx}{\int \left[w(x) f_X(x|C_1)^2\right] f_X(x) dx} \tag{3.9}$$

The integration is performed over the selected region. In computing eq. (3.9), the expectation is replaced by the ensemble average over the x's in the given region. Note that due to taking expectation, the squared error between $Nf_X(x)$ and $N_1 f_X(x|C_1)$ in eq. (3.8.d) is in fact weighted according to the density function of $f_X(x)$. If the least square error is desired, this weight can be canceled out by using the weight function $w(x) = 1/f_X(x)$. This matching process can also be applied at the level of the cumulative distribution function and this can be achieved with the error function in eq. (3.8.d) by employing the following weight function which further cancels out the effect of $f_X(x|C_1)$.

$$w(x) = \frac{1}{f_x(x|C_1)f_x(x)}$$

Then, the estimated value N_1 is computed as,

$$\hat{N}_1 = \frac{\int Nf_x(x)\,dx}{\int f_x(x|C_1)\,dx} \tag{3.10.a}$$

The numerator in eq. (3.10.a) is the expected number of samples found in the selected region. Since the samples of C_1 are assumed to dominate in their numbers over the samples from the "others" class in the given region, the numerator in eq. (3.10.a) will be the approximated number of samples from C_1. Suppose the integrated value of the denominator is a,

$$\alpha = \int f_x(x|C_1)\,dx \tag{3.10.b}$$

which is the probability of class C_1 for the given region. By performing the significance testing with the acceptance probability a, the estimate of N_1 in eq. (3.1O.a) can be easily obtained by counting the number of samples accepted and dividing by 'the selected acceptance probability a. The estimate in eq. (3.1O.a) will be in most cases an over-estimated value, since there should be samples not belonging to class C_1 in the count of the numerator in eq. (3.110.a). This over-estimation will be significant, especially when there is insufficient separability between the class of interest and the class of "others." In developing the initial clusters specification, experimental results show that this over-estimation is not critical to the performance, but an under-estimated value could be problematic since it results in non-trivial $\overline{w}_{i1}$ values and causes clusters generated in the region where most of the class-of-interest samples are located. These extraneous clusters would take a significant portions of class C_1 samples away.

In the experiment, eq. (3.10.a) is used in the one dimensional y space where $y = x^Tx$ due to its simplicity. Note that this matching can be computationally burdensome unless the matching is processed in a lower dimensional space, such as the one-dimensional y space.

For those cases when the region for the matching process cannot be easily selected, a slightly different algorithm is developed. Note that, using the weight w_{i1}'s in eq. (3.2.b), the probability density function $f_X(x|C_1)$ can be estimated from the unlabeled samples with weights. For example, under the Gaussian assumption of $f_X(x|C_1)$, the mean and covariance matrix of it (denoted as $M(\pi_1)$ and $\Sigma(\pi_1)$, respectively) can be estimated as,

$$M(\pi_1) = \frac{1}{W_1}\sum_{i=1}^{N} w_{i1}\, x_i \tag{3.11.a}$$

$$\Sigma(\pi_1) = \frac{1}{W_1}\sum_{i=1}^{N} w_{i1}\, (x_i - M)(x_i - M)^T \tag{3.11.b}$$

where W_1 is the sum of weights w_{i1}'s and is computed as,

$$W_1 = \sum_{i=1}^{N} w_{i1}$$

and the function $h(x \mid \pi_1)$ is the Gaussian density function with mean and covariance matrix, $M(\pi_1)$ and $\Sigma(\pi_1)$. $h(x \mid \pi_1)$ will be compared with the function $h(x)$ while varying π_1. This is a recursive process since the best value of the prior π_1 is found by checking the value π_1 with which the estimated $h(x \mid \pi_1)$ is most similar to $h(x)$. Note that this is based on the assumption that the nearer to the true value the unknown π_1 is, the more $h(x \mid \pi_1)$ match well with $h(x)$.

For comparison of the two functions, any statistical separability measure, such as the divergence, the Jeffries-Matusita (JM) distance, or the transformed divergence (Swain 78) can be used to quantify the similarity. This procedure doesn't require specifying the region over which the matching should take place. Note that, at least in principle, this procedure is not limited to the parametric case, although the computation required in estimating recursively $h(x \mid \pi_1)$ and evaluating the similarity measure in a non-parametric case may be formidable. In computing the weights w_{i1}'s, the simple procedure in eq. (3.3) in previous section can be used.

3.4 Experiments and Discussion

To test the performance of the partially supervised classification algorithm proposed in this chapter, experiments were carried out with both simulated and real data. The partially supervised classification algorithm should be effective even when the class of interest is not well separated from the others. To test the proposed algorithm over a wide range of separability, several bivariate Gaussian data sets were generated with different degrees of separability as in the previous chapter. In the case of real data, the July LANDSAT Thematic Mapper (TM) data introduced in previous Chapter were used. For comparison, the (relative) maximum likelihood classifier (denoted as "REL-ML") was designed with the known class statistics, and the classification error was used for evaluation.

3.4.1 Experiments and Discussion

For a test with simulated data, as in previous chapter, bivariate ($q = 2$) Gaussian data were generated. 1000 samples were generated for the class of interest, C_{int} with zero mean and an identity covariance matrix. The class "others," C_{others}, was assumed to be Gaussian with mean $[d,0]^T$, $d > 0$, and an identity covariance matrix. 2000 samples were generated for C_{others}.

To avoid any random error due to the data generation process and its effect on evaluating experimental results, data sets were generated 50 times with different seed numbers and the averaged result used in comparison.

Equation (3.10.a) was used to obtain the N, estimate with varying acceptance probability, a, in eq. (3.10.b) as in Fig. 3.4. The estimated values were not much different for different a's. As expected, unless the separability between the two classes is sufficient, there was significant over-estimation.

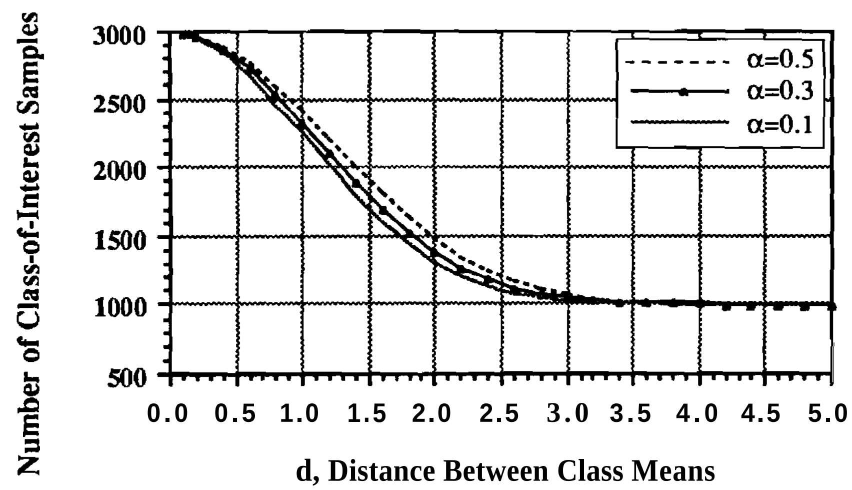


Figure 3.4 Estimated Number of Class-of-Interest Samples with Different Values of Acceptance Probability a's Using eq. (3.10.a, b).

Using the N_1 estimate, the weights $\overline{w}_{i1}$'s were computed and used in unsupervised clustering to develop clusters corresponding to the "others" class. Any cluster which had a negligible effective number from eq. (3.6.a), or a negligible ratio from eq. (3.6.b) was deleted. Without the ratio checking, due to non-trivial weights $\overline{w}_{i1}$'s in the regions where the weights should be negligible, an under-estimated value of N_1 would result in extraneous clusters and cause large omission error. For those clusters, the effective numbers of samples in eq. (3.6.a) would be much smaller than the actual sample numbers grouped to those clusters since significant portions of the samples in those clusters are from the class of interest. Those extraneous clusters can also be observed even though N_1 is not much under-estimated in such cases when the actual distribution of the class-of-interest samples is slightly different from that predicted by the probability density function $f_X(x|C_1)$.

Figure 3.5 shows the class-averaged classification error comparisons of the relative maximum likelihood classifier (denoted as "REL-ML"), a partially supervised classifier based on significance testing (denoted as "ABS-SIG") and the proposed classifier based on unsupervised clustering with three different acceptance probability a's for the N_1 estimation (denoted with three different a values).

The class-averaged classification error is a simple average of the omission and commission errors. The result of the significance testing is obtained by scanning the significance level in the interval [0.01, 0.991 in steps of 0.01, choosing the best one. Therefore, this is the best one attainable with significance testing. While significance testing had about 5 ~ 10% greater error than the relative maximum likelihood classifier unless the distance d between two class means was sufficiently large, the proposed algorithm closely followed the performance of the maximum likelihood classifier. Only when the overlap between two classes is significant (see the case d < 2, for example) and the N_1 value iis severely over-estimated, was there as much as 5% error increase compared to the maximum likelihood classifier.

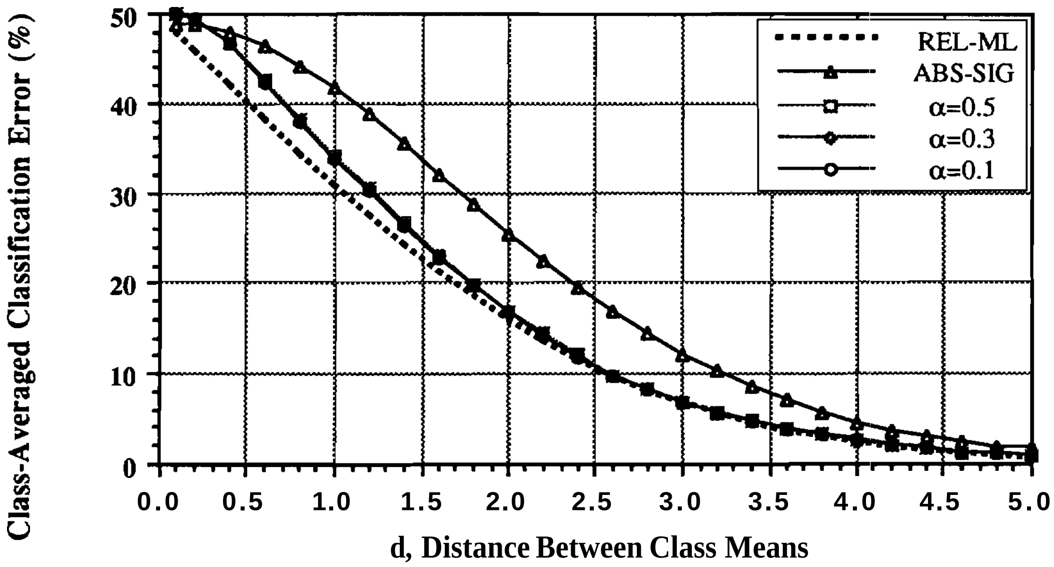


Figure 3.5 Class-Averaged Classification Error Comparison.; The proposed classifier based on unsupervised clustering is denoted by the a value of eq. (3.10.b) used in estimating the number of class-of-interest samples. "REL-ML" is the relative maximum likelihood classifier with known class statistics, and "ABS-SIG" is the best result for significance testing attainable with significance levels in the interval [0.01, 0.99].

To see the sensitivity of the proposed classification algorithm to the N_1 estimate and its amount of under- or over-estimation, several different values of N_1 were used in computing the weights $\overline{w}_{i1}$'s in the clustering without estimating it. The classification result is shown in Fig. 3.6.

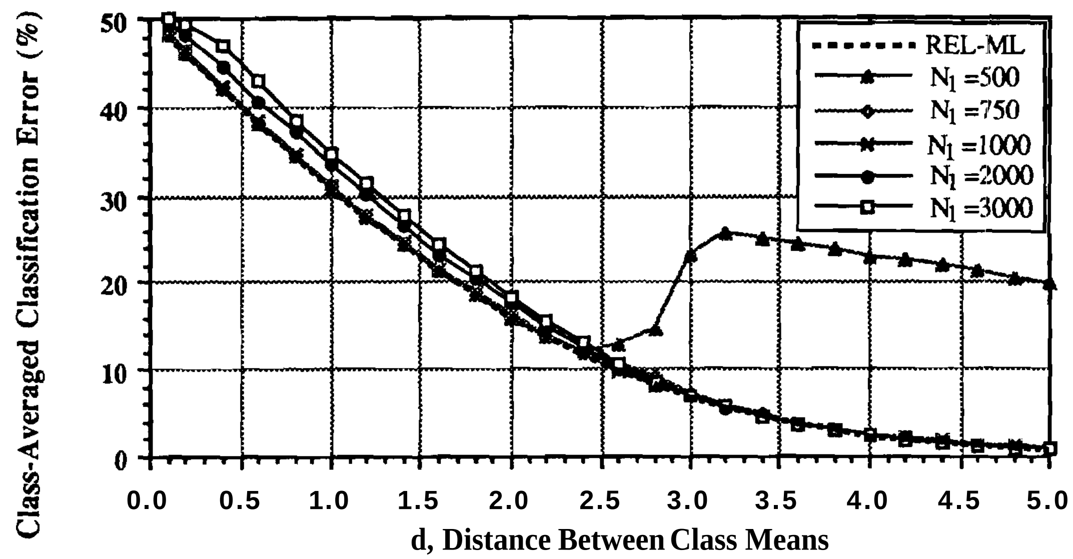


Figure 3.6 Sensitivity on the Estimate N, of the Proposed Classifier.; Several different values of N, were used in computing the weights $\overline{w}_{i1}$'s in the clustering without estimating.

There was almost negligible difference in class-averaged classification error when N, was varied from 750 to 1500 (not shown). When an over-estimated N_1 was used, there was as much as 2% (N_1 = 2000, 100% over-estimation) or 5% (N, = 3000, 200% over-estimation) error increase compared to the maximum likelihood classifier when d < 2. An over-estimated value of N_1 increases the commission error and its effect becomes more noticeable as the overlap between two classes increases. Although the proposed algorithm was very tolerable of the degree of over-estimation, it was less so with under-estimation as shown for the case N_1 = 500 (50% under-estimation) in Fig. 3.6. When d > 2.5, the class-averaged classification error increased since the clusters containing a non-trivial portion of the class-of-interest samples survived the cluster deletion test of eq. (3.6.a,b) and many class-of-interest samples were deleted to increase omission error. Note that, as shown in Fig. 3.4, the N_1 estimate with eq. (3.10.a,b) is in general slightly over-estimated due to the commission of "others" samples in the numerator of eq. (3.1O.a). Therefore the under-estimation is not so problematic unless the training samples of the class of interest are not representative enough to adequately model its distribution function.

3.4.2 Experiment with Thematic Mapper data

For a test with real data, the July LANDSAT Thematic Mapper(TM) data which was also used in the previous chapter was used. For a description of training and test samples, refer to Table 2.2.

For comparison of classification performance, a maximum likelihood classifier was designed with all 12 sub-classes and the classification errors were evaluated. The performance of the classifier was assessed in terms of class-averaged classification error, total classification error and a simple average of these two. As discussed in the previous chapter, note that while the class-averaged classification error is a simple average of the omission and commission errors, the total classification error is a weighted average of those two errors according to the prior probabilities of the class of interest and the others.

Classification was also performed with significance testing and the proposed algorithm based on unsupervised clustering. Since there are more than one sub-class for each information class, the whole data set was first divided using a maximum likelihood classifier into n sub-groups where n is the number of sub-classes of a given information class. For each sub-group, the proposed classifier was applied to identify the samples belonging to the corresponding sub-class.

Figure 3.7 shows the classification error comparison of significance testing and the proposed partially supervised classifier based on unsupervised clustering. As before, various values were tried to find the best significance level for each sub-class. When estimating N, in the proposed classifier, five different values of a (0.1, 0.2, 0.3, 0.4 and 0.5) were used and the estimated numbers;N_1 were mostly over-estimated. Since less than 1% of the differences are observed in the classification error even though there were large differences in the degree of over-estimation (21% ~ 177%), only the result with a = 0.9 is shown in Fig. 3.7. The proposed algorithm is seen to perform better in all classes by about 1 ~ 6% than the ***best*** significance testing case where the significance levels were deliberately chosen manually.

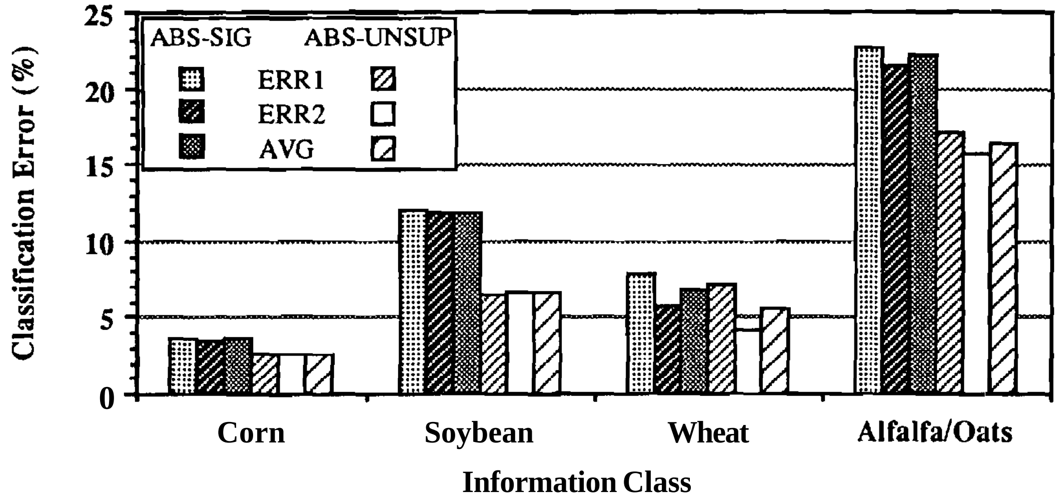


Figure 3.7 Classification Error Comparison of Significance Testing (ABS-SIG) and the Proposed Classifier Based on Unsupervised Clustering (ABS-UNSUP).; N, was estimated with α=0.9. The comparison is made with class-averaged classification error (denoted as "ERR1"), total classification error (denoted as, "ERR2") and the simple average of those two (denoted as "AVG").

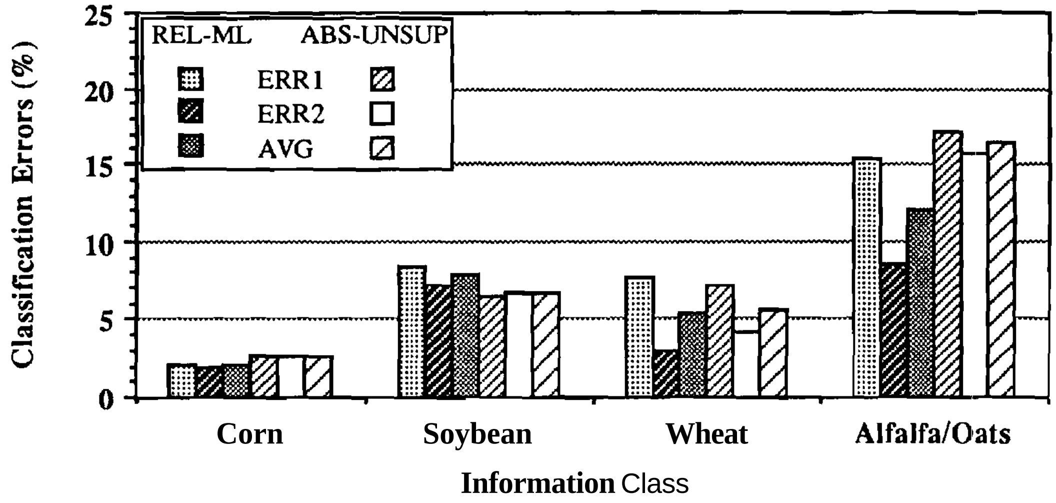


Figure 3.8 Classification Error Comparison of the Proposed Classifier (ABS-UNSUP) with the Relative Maximum Likelihood Classifier (REL-ML).; The comparison is made with the class-averaged classification error (denoted as "ERR1"), the total classification error (denoted as, "ERR2") and the simple average of those two (denoted as "AVG").

In Fig. 3.8, the classification error comparison is made with the relative maximum likelihood classifier. Except for the class, "alfalfa/oats", there was only 1 ~ 2% difference in classification error compared to the relative maximum likelihood classifier. As for the class "alfalfa/oats", there was about 7% increase in commission error compared to the maximum likelihood classifier.

Figures 3.9 to 3.12 show the locations of the samples identified by the proposed partially supervised classifiers and the relative maximum likelihood classifier.

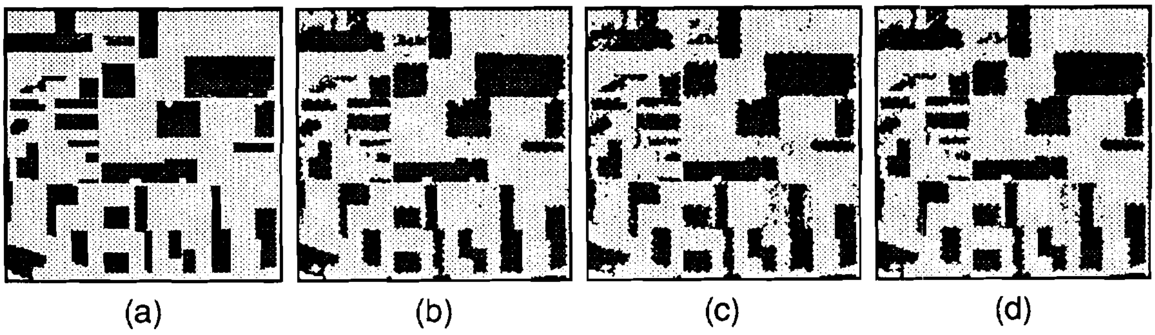


Figure 3.9 Results For the Class "Corn" Samples. (a) Ground truth location of corn samples. (b) Result for the relative maximum likelihood classifier (REL-ML). (c) Result for the best significance!testing (ABS-SIG). (d) Result for the unsupervised clustering based proposed classification (ABS-UNSUP).

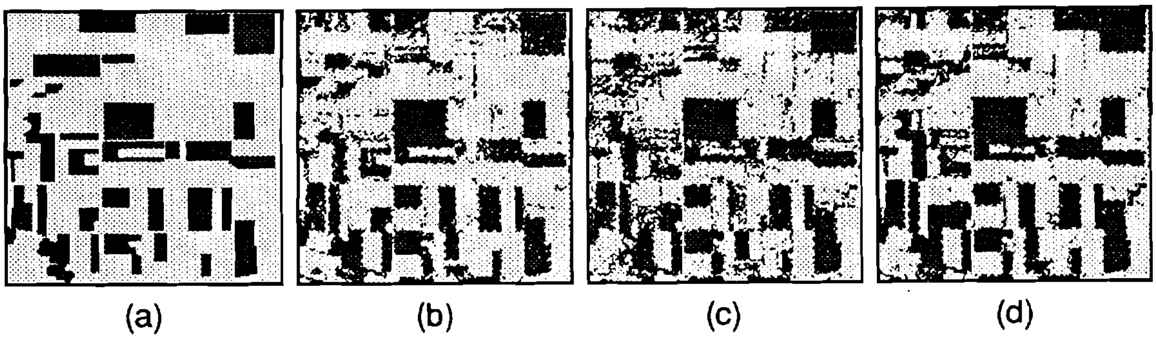


Figure 3.10 Results for the Class "Soybeans" Samples. (a) Grouncl truth location of soybean samples. (b) Result for the relative maximum likelihood classifier (REL-ML). (c) Result for the best significance testing (ABS-SIG). (d) Result for the unsupervised clustering based proposed classifier (ABS-UNSUP).

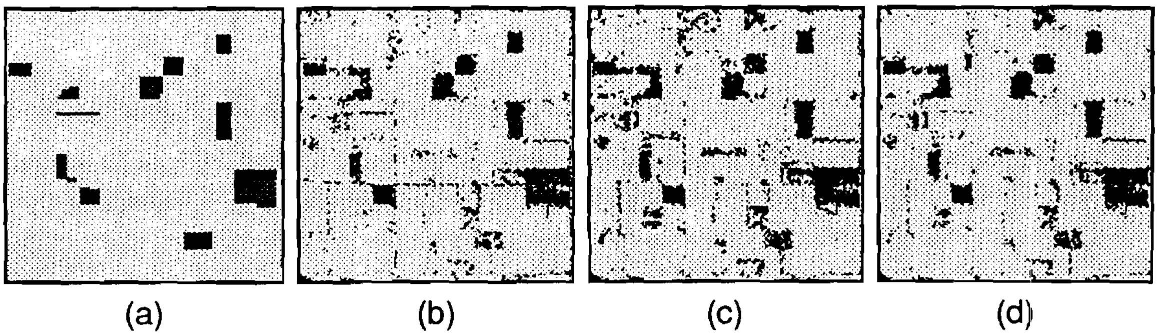


Figure 3.11 Results for the Class "Wheat" Samples. (a) Ground truth location of wheat samples. (b) Result for the relative maximum likelihood classifier (REL-ML). (c) Result for the best significance testing (ABS-SIG). (d) Result for the unsupervised clustering based proposed classifier (ABS-UNSUP).

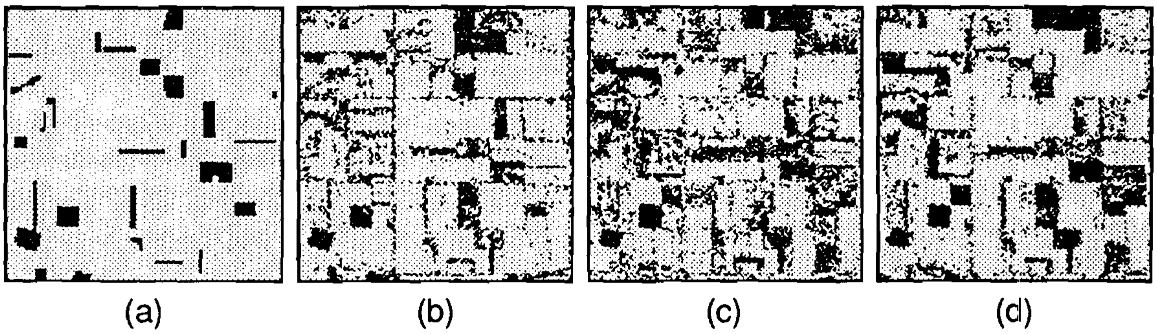


Figure 3.12 Results for the Class "Alfalfa/Oats" Samples. (a) Ground truth location of alfalfa/oat samples. (b) Result for the relative maximum likelihood classifier (REL-ML). (c) Result for the best significance testing (ABS-SIG). (d) Result for the unsupervised clustering based proposed classifier (ABS-UNSUP).

Compared to the relative maximum likelihood classifier, which requires a complete list of classes to be defined and their class statistics computed, the proposed classification algorithm was very successful in its classification performance even though prior knowledge was provided only for the specific information class under consideration. The computational complexity was increased over the relative maximum likelihood classifier, but not prohibitively so in view of the time savings for the manual portion of the analysis task. In the experiment with Thematic Mapper data in identifying one information class, it took on average about 3 times more computational time than the maximum likelihood classifier.

3.5 Summary

In this chapter, we have proposed a partially supervised classification algorithm based on unsupervised clustering. Initiated with only prior knowledge pertaining to a particular class to be identified, the proposed classifier develops class statistics of "others" class through a weighted unsupervised clustering procedure. The user only needs to provide the information for a particular class one actually wants to identify.

Experiments with both simulated and real Thematic Mapper data showed very satisfactory classification performance compared to the standard relative maximum likelihood classifier. The proposed classification algorithm is also computationally moderated compared to the maximum likelihood classifier. The operational simplicity should make this algorithm useful in many practical applications.

3.6 Conclusions of the Partially Supervised classification and Suggestions for Future Research

In Chapter 2 and 3, the problem of partially supervised classification especially when the class definition and corresponding class statistics are available a *priori* only for a particular class of interest. This problem can be frequently encountered in many real application of pattern classification techniques. Two approaches, one being based on significance testing, which belongs to the absolute classification scheme, and the other being based on the unsupervised clustering, belonging to the relative classification scheme, were proposed.

The experiments both with simulated and real LANDSAT Thematic Mapper data showed very satisfactory results compared to the maximum likelihood classifier which was designed with complete prior knowledge.

The optimal acceptance probabilities estimated without supervision for significance testing matched very well with the manually selected optimal values. Significance testing inherently has a limitation in its classification performance due to the dimensional reduction of the feature space. This effect was noticeable in the experiments. However, the second approach based on unsupervised clustering doesn't have this limitation since it performs classification in the original feature space without dimensionality reduction. But this requires the knowledge of a number of class-of-interest samples in the given data set. The simple procedure based on thresholding and counting numbers of samples accepted with the given thresholding was found to be satisfactory for initiating the unsupervised clustering to find the clusters corresponding to the unknown class of "others."

However, there are needs for deciding *a priori* various parameter values which control the clustering procedure. For the proposed algorithm based on clustering to be fully and easily usable by users with little prior knowledge about the data set, there must be a dependable algorithm which can suggest at least a proper range of parameter values for clustering. These parameter values are expected to be also dependent on the particular clustering algorithm selected. In fact, this is very closely related to the general clustering problem.

In designing a partially supervised classifier, the quality of training samples is of utmost importance. To properly design the classifier, the training samples must be representative of the same class samples in the given data set to be classified. To fulfill this requirement in the previous experiments, random sampling was carried out to sample about 10% of data from the data set to use as training samples and the resultant randomly selected training samples were found to be satisfactorily representative. However, in a practical application of the classifier, the limited training samples won't be always very representative all the time. In a conventional *totally supervised* relative classification, there is somewhat of a wide tolerance for the representativity requirement, due to its *relative* consideration in decision making, but, in the partially supervised classification case, there is expected to be less tolerance. Therefore, developing a robust partially supervised classifier will be very important.

LIST OF REFERENCES

Bello, M. G., "A Random-Field Model-Based Algorithm for Anomalous Complex Image Pixel Detection," IEEE Trans. Image Processing, Vol. 1, No. 2, pp.186-196, April, 1992

Boneva, L. I., D. G. Kendall and I. Stefanov, "Spline transformations : three new diagnostic aids for the statistical data-analyst (with Discussions)," Journal Royal Statist. Soc. B, 33, pp.1-70, 1971

Dempster, A. P., N. M. Laird and D. B. Rubin, Maximum Likelihood from Incomplete Data Via the EM Algorithm (with discussions)," Journal of Royal Statist. Soc. B, 39, pp.1-38, 1977

Drake, A., Fundamentals of Applied Probability Theory, McGraw-Hill, New York, 1967

Fukunaga, K., Introduction to Statistical Pattern Recognition, 2nd Edition, Academic Press, New York, 1990

Fukunaga, K., R. R. Hayes, and L. M. Novak, "The Acquisition Probability for a Minimum Distance One-class Classifier," Trans. IEEE Aerospace and Electronic Systems, AES-23, pp.493-499, 1987

Hall, D. J. and G. B. Ball,"ISODATA : A novel method of data analysis and pattern classification," Technical Report, Stanford Research Institute, Menlo Park, CA, 1965

Hoffer, R. M., M. D. Fleming, L.A. Bartolucci, S. M. Davis, R. F. Nelson, "Digital Processing of Landsat MSS and Topographic Data to Improve Capabilities for Computerized Mapping of Forest Cover Types," LARS Technical Report 011579, Purdue University, West Lafayette, IN 47907, 1979

Quatieri, T. F., "Object Detection by two-dimensional linear precliction," Proc. of IEEE Intern. Conf. on Acoustics, Speech and Signal Processing, ICASSP 83, pp.108-111, April, 1983

Redner, R.A. and H. F. Walker, "Mixture densities, maximum likelihood and the EM Algorithm," SIAM Rev., 26, pp.195 - 239, 1984

Richards J. A., "Remote Sensing Digital Image Analysis, An Introduction," Spring-Verlag, Berlin, Heidelberg, 1986

Silverman, B. W., Density Estimation for Statistics and Data Analysis, Chapman and Hall, 1986

Swain, P. H., "Fundamentals of Pattern Recognition in Remote Sensing," Remote Sensing - The Quantitative Approach, edited by P. H. Swain and S. Davis, McGraw-Hill Book Company, New York, 1978

Therrien, C. W., T. F. Quatieri, and D. E. Dudgeon, "Statistical Model-Based Algorithms for Image Analysis," Proceedings of IEEE, Vol. 74, No. 4, pp.532-551, April, 1986

Titterington, M., *et al.*, Statistical Analvsis of Finite Mixture Distributions, John Wiley & Sons, New York, 1985

Van Trees, H. L., Detection. Estimation. and Modulation Theorv. Part I, John Wiley & Sons, New York, 1968

APPENDIX A

Fast Parzen Density Estimate Using Clustering-Based Branch and Bound

A.l Introduction

Applying statistical pattern recognition techniques often requires the probability density functions of given data samples. If the distribution of the given data can be assumed to follow a certain known parametric form, such as a Gaussian distribution, then, the parameters specifying the density function can be estimated. However, it is not always possible to assume a certain parametric distribution function for the given data set without causing significant error. In this case, a non-parametric approach can be taken by employing a density estimation technique [A.1]. Since the process of density estimation usually takes substantial computation, it might not be feasible to adopt this non-parametric approach, especially in an on-line application. There has been research on reducing the computational requirement of the density estimation based on k nearest neighbor [A.2,3] by saving the number of evaluations of quadratic terms which are required to find the k nearest neighbors. As for the Parzen density estimate, there has also been research on selecting a representative subset of the given training data set [A.4]. The reduced subset of training samples are selected in such a way that the Parzen density estimate with the reduced set matches as closely as possible with that with full data set in the sense of the entropy measure of similarity between two estimates. If the Parzen density estimate is to be evaluated on a regular grids, for example, in plotting the density function or drawing a contour diagram, the fast Fourier transform (FFT) can be used by noting that the Parzen density estimate is the convolution of the data with the kernel function [A.5]. In the general case of evaluating at irregular points, this algorithm is not applicable.

In this appendix, similarly to the efficient density estimate based on k nearest neighbor [A.2,3], the branch-and-bound procedure is applied in Parzen density estimation to reduce the number of evaluations of quadratic terms. Noting that the contribution of a training (or design) sample on the evaluated density estimate rapidly diminishes if it is far away from the location of evaluation, therefore, without causing much error, some of the training samples could be left out in evaluating the kernel functions if the distances from the location of evaluation to those samples exceed a certain critical distance. The computation required for checking the distances can be significantly reduced by utilizing the branch-and-bound procedure. Experimental results are presented to show the effectiveness of the proposed approach in reducing the computational load on the Parzen density estimation. Notice that, to further reduce the computational burden, this proposed algorithm also can be used in addition to the data reduction algorithm in [A.4].

A.2 Fast Parzen Density Estimation

Suppose there is a training data set, Y with N elements from which the unknown density function should be estimated. The dimensionality of the data is denoted by q ($q \geq 1$). The q-dimensional feature space is indicated by R^q. The Parzen density estimate $\hat{f}_X(x)$ of the unknown probability density function at x, $x \in R^q$, is obtained as a sum of kernel functions placed at each sample y in Y as,

$$\hat{f}_X(x) = \frac{1}{N h^q} \sum_{y \in Y} K\left[\frac{x-y}{h} \right] \qquad \text{(A.1.a)}$$

where $K(\bullet)$ is the selected kernel function and h is the smoothing parameter (or, window size). The kernel function satisfies the following condition,

$$\int_{x \in R^q} K(x)\, dx = 1 \qquad \text{(A.1.b)}$$

Since the estimated density $\hat{f}_X(x)$ will inherit all the properties of the selected kernel function, the kernel function is often chosen in such a way that it has mathematically tractable properties such as continuity or differentiability. Some examples include the Gaussian kernel function, Epanechnikov kernel function, or the rectangular kernel function [A.1]. The value of the kernel function rapidly decreases as the distance from the origin increases. Therefore, the contribution of a sample in Y to the estimated probability density at a certain x will become negligible if the distance between x and the sample in Y becomes large. Without introducing significant error, in many situations, it is possible to select a "critical distance", D_c and to assume the contribution of a sample y in Y is negligible if the distance to x is more than this critical distance. In estimating a density, a truncated and rescaled version of the original kernel function is used to satisfy the condition in eq. (A.1.b). Suppose the truncation level is denoted by β, then, the critical distance D_C with window size h=1, is computed as,

$$\beta = \int_{x^T x \leq D_c^2} K(x)\, dx \ , \qquad 0 \leq \beta \leq 1 \qquad \text{(A.2.a)}$$

The critical distance with window size h is then obtained by multiplying h with the D_C calculated in eq. (A.2.a). The truncated kernel function with truncation level β is denoted by $K'(x; \beta)$ and given as,

$$K'(x; \beta) = \frac{K(x)}{\beta} \qquad x^T x \leq D_c^2$$

$$= 0 \qquad \text{otherwise} \qquad \text{(A.2.b)}$$

Depending on the specific application and the degree of permissible trade-off between accuracy and speed, an appropriate value of β in eq. (A.2.a) can be selected. Some kernel functions such as the Epanechnikov kernel function or the rectangular kernel function have compact support in the given feature space only on which the function has non zero values. In these cases, it is straightforward to select the value D_C without losing any accuracy, and there is no need for truncation and normalization.

Denote the distance between two samples, x and y as L(x, y). If the Euclidean distance measure is used, then, L(x, y) is computed as,

$$L(x, y) = \sqrt{(x-y)^T(x-y)} \qquad \text{(A.3.a)}$$

If different smoothing parameters are to be allowed for different coordinate directions, then, a slightly modified measure of distance can be used with the kernel covariance matrix Σ,

$$L(x, y) = \sqrt{(x-y)^T\, \Sigma^{-1}\, (x-y)} \qquad \text{(A.3.b)}$$

Note that the distance measure in eq. (A.3.b) is equivalent to the Euclidean distance measure in eq. (A.3.a) after pre-whitening [A.6] with the appropriate Σ. Pre-whitening is assumed to be already performed, if required, to the data Y and x's to deal with the need of different smoothing parameters, and in the subsequent discussion, the Euclidean distance measure will be used.

Suppose the Parzen density estimate is evaluated at x. Notice that a sample y in Y which doesn't satisfy,

$$L(x, y) < D_c \qquad \text{(A.4)}$$

can be excluded from the computation in eq. (A.l). The number of checking distances in eq. (A.4) can be significantly reduced by using the critical distance D_c and applying the branch and bound algorithm [A.3] with clustering.

Suppose clustering is performed to group the samples in Y into clusters. To each cluster, for example, to the j^{th} cluster C_j, three variables, $\{I_j, M_j, D_{max}(j)\}$ are associated. M_j is the cluster mean and I_j is the index set of cluster C_j defined as,

$$I_j \equiv \{i \mid i^{th} \text{ sample } y_i \text{ belongs to cluster } C_j,\ y_i \in Y\}$$

$$D_{max}(j) \quad \max_{i \in I_j} \{L(x_i, M_j)\}$$

$D_{max}(j)$ denotes the maximum distance from the cluster mean, M_j to the samples in cluster C_j. Notice that the distance from x to any sample in C_j should be larger than $L(x, M_j) - D_{max}(j)$. Therefore all the samples belonging to the cluster C_j which don't satisfy the inequality in eq. (A.5) can be excluded in evaluating the density estimate at x as shown in Figure A.1.

$$L(x, M_j) - D_{max}(j) < D_c \qquad \text{(A.5)}$$

Therefore, the calculation of distances from x to each sample in Y can be significantly reduced by checking this inequality and deleting clusters appropriately. Note that this same idea can be also applied to reduce the number of clusters which need be checked with this inequality by creating a hierarchical grouping of the clusters, but we will not elaborate here.

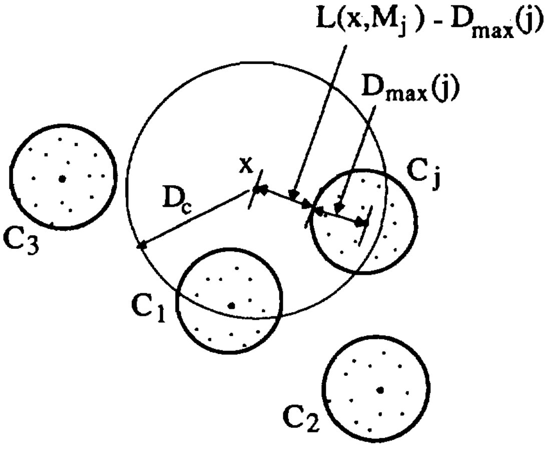


Figure A.l **Efficient Computation of Parzen Density Estimate Using Clustering.; Samples grouped into clusters other than C_2 and C_3 in this figure need not be considered in the computation of Parzen density estimate.**

The computation required for the clustering may be not trivial, but it is required only once for each training data set. If the number of locations for which the probability density should be computed is large, then this one-time extra computation for clustering should be worthwhile. When the probability density is actually evaluated, there exists another extra computation required for the distances from given location x to each cluster center. Considering the savings due to skipping a subset of distant training samples, this will be quite negligible unless the number of clusters is comparable to the number of total training samples.

In unsupervised clustering, a new cluster is generated if the minimum distance to the existing clusters exceeds the pre-specified distance (let us denote this by T_{create}). To achieve a maximal efficiency in reducing computational load, care must be exercised in selecting a proper value of T_{create}. Too small a value of T_{create} will result in a large number of small clusters into which very small numbers of samples are grouped. In this case, the overhead of clustering and checking the inequality in eq. (A.5) will surpass the savings obtained by skipping the samples grouped in distant clusters. On the other hand, a small number of large clusters due to too large a value of T_{create} might not be able to eliminate any clusters in evaluating the density estimate. The value of T_{create} should be related to the critical distance D_c

A.3 Experiments and Discussion

To verify the effectiveness of the fast Parzen density estimation algorithm proposed, an experiment with simulated data was performed. For a training data set, 1000 samples of bivariate (q = 2) normal data were generated. The mean and covariance matrix were set to $[0, 0]^T$ and the identity matrix, respectively. The density estimate was evaluated at four different groups of locations. That is, four sets of bivariate Gaussian samples, each containing 100 samples, were generated with means at $[\pm 1.5, 0]^T$ and $[0, \pm 1.5]^T$. The covariance matrices were all set to the identify matrix.

To see the effect on the efficiency of this algorithm, the parameter for new cluster generation, T_{create}, was selected as,

$$T_{create} = \alpha D_C \tag{A.6}$$

and the value a was varied to see its effect on the effectiveness of the proposed algorithm. (In clustering, if the squared distance to the nearest cluster is more than $9 T^2_{create}$, then a new cluster is generated. Therefore, the maximum distance $D_{max}(\cdot)$ in eq. (A.5) is $\sqrt{q}\, T_{create}$). The effectiveness of this algorithm was measured in terms of percent of the number of distance computations actually evaluated in density estimation.

$$R \equiv 100 \text{ x } \frac{\text{average number of quadratic distance computation}}{\text{number of training samples}} \tag{A.7}$$

In the numerator in eq. (A.7), the number of distance computations to the cluster centers is also included even though it might be negligible in most cases. The averaging is carried out for the test samples. In the case of conventional Parzen density estimation, the percent ratio R in eq. (A.7) is 100. If the overhead of computing distances to the cluster centers surpasses the savings acquired by deleting some of the distant clusters, the ratio R can be greater than 100.

First, the Epanechnikov kernel function was used since it is straightforward to choose the critical distance D_C, which is equal to the window size h. As suggested in (p.86 in [A.1]), the window size h was set to 0.56 in the case of this Epanechnikov kernel function. Under this setting, only 4.24% of the training samples on the average actually contributed in the density estimation. The value a in eq. (A.6) was varied from 0.01 to 8 to see the effect of the numbers of clusters on deleting some of the distant clusters. Only one iteration of clustering was performed since a crude grouping of the samples is sufficient. As a in eq. (A.6) decreases (in other words, as the number of clusters increases), the savings in distance computation increases up to a certain point, and after which the overhead of distance computation to the cluster centers overwhelms the savings attained by skipping some of the training samples as seen in Fig. A.2.

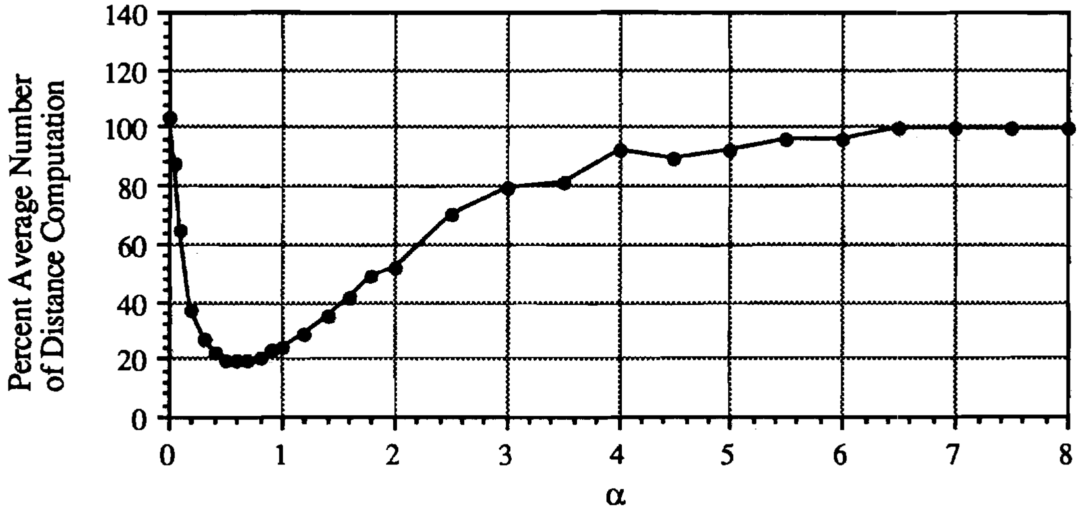


Figure A.2 R, Average Number of Distance Computations in eq. (A.7) Expressed as a Percent.; The Epanechnikov kernel function was used with different cluster creation conditions as in eq. (A.6) where a was varied from 0.01 to 8.; window size h = 0.56, and critical distance D_C = 0.56.

Unless a is extremely small (unless a < 0.02 in this experiment), the overhead was negligible. About 80% of the savings was observed in distance computation with the value a in 0.5 ~ 1.0.

The same experiment was performed with the Gaussian kernel function, which has non-zero values in the entire feature space. The truncation was performed with truncation level β as in eq. (A.2. a & b). The window size was set to h=0.304 as suggested in (p.86 in [A.1]). The truncation level β was varied from 0.8 ~ 0.999. Notice that there are some training samples which do not make any contribution in the density estimate even without using the truncated kernel function. In other words, due to the numerically finite precision, the value of the exponential function in Gaussian kernel function becomes (numerical) zero if its argument is too small.

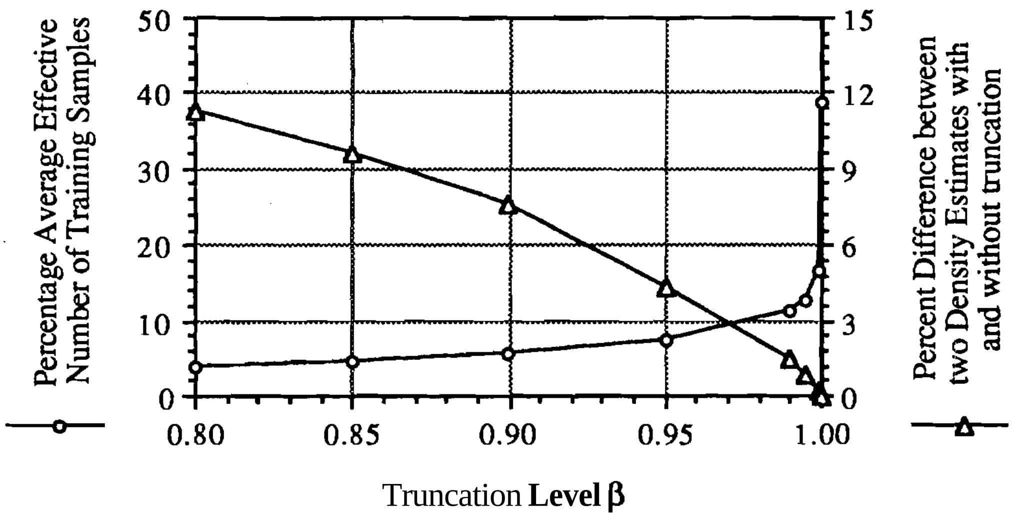


Figure A.3 Percent Average Effective Number of Training Samples versus Truncation Level.; This shows the percent average effective number of training samples which have non-zero contribution to the density estimate and the corresponding average percent difference between density estimates obtained with and without truncation.; The Truncated Gaussian kernel function was used with different truncation level β's.; the window size was set to h = 0.304.

Figure A.3 shows the average number of effective training samples which give non-zero values of the exponential function when a truncated Gaussian kernel function with truncation level β is used. The number obtained without truncation is considered as that of β=1.0. There must be error introduced due to the truncation of the kernel function and the amount of error is measured by the average percent difference between the two density estimates obtained with and without truncation as,

$$\text{Average percent difference} = 100 \times E_x \frac{\left|\hat{f}_X(x;\beta) - \hat{f}_X(x)\right|}{\hat{f}_X(x)} \qquad \text{(A.8)}$$

where $\hat{f}_X(x)$ denotes the density estimate without truncation and $\hat{f}_X(x;\beta)$ denotes the density estimate with the truncation level set to β. The expectation in eq. (A.8) is obtained by computing the mean over the given 40Q test samples. As seen in Fig. A.3, even when β = 1.0, there were only 38.64% of the training samples which actually contributed to the density estimate due to the numerically finite precision. When β = 0.999, the effective number of training samples dropped to 16.46%, but there was only 0.19% difference to the average between $\hat{f}_X(x)$ and $\hat{f}_X(x;\beta)$. If β = 0.99, the percent difference was 1.47% with 11.17% of the effective training samples. Whether or not this error due to truncation is acceptable depends on each particular application of the estimated density in mind.

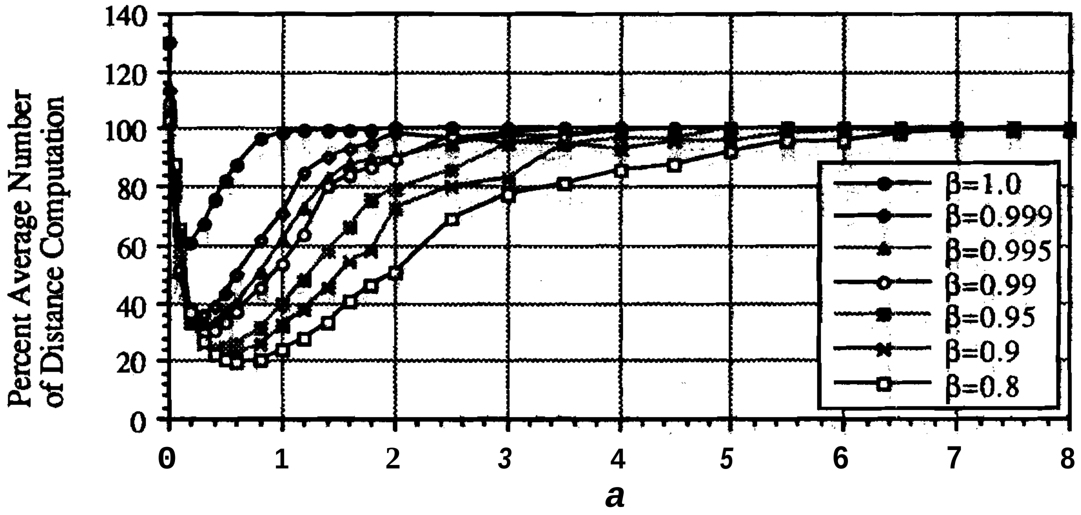


Figure A.4 Percent Average Number of Distance Computation R.; Truncated Gaussian kernel function with truncation level β. The parameter a in the cluster creation condition of eq. (A.6) was varied from 0.01 to 8.; window size h = 0.304.

As before, while the parameter a in eq. (A.6) is varied from 0.01 to 8, the average number of actual distance computations is shown in Fig. A.4. As the truncation level β becomes larger, the amount savings in distance computation increases. In both Fig. A.3 and A.4, very small or very large values of a were not acceptable, since they produce too many small clusters or just one or two large clusters. With a in the range of 0.2 ~ 1.0, it is observed that about 40 ~ 80% savings in distance computation can be achieved.

A.4 Conclusions

In this appendix, a computationally efficient Parzen density estimation algorithm is developed by adopting the idea of the branch and bound method with clustering. Not only those kernel functions having finite support for non-zero values such as the Epanechnikov kernel function, but also the kernel functions having non-zero values over the entire feature space was applicable with this algorithm with truncation. By choosing a proper parameter setting for D_c for new cluster generation, the savings in computation is observed to be maximized. The experimental results verified significant savings in computation.

A.5 List of References for Appendix A

[A.1] B. W. Silverman, Densitv Estimation for Statistics and Data Analvsis, Chapman and Hall, 1986

[A.2] P. E. Hart, "The condensed nearest neighbor rule," IEEE Trans. Information Theory, IT-14, pp.515-516, 1968

[A.3] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," IEEE Trans. Computers, C-24, pp.750-753,1975

[A.4] K. Fukunaga and R. R. Hayes, "The reduced Parzen classifier," IEEE Trans. Pattern Anal. and Machine Intell., PAMI-11, pp.423-425, 1989

[A.5] B. W. Silverman, "Kernel density estimation using the fast Fourier transform," Statistical Algorithm, AS176, Appl. Statist. 31, pp.93-97, 1982

APPENDIX B Program List for Partially Supervised Classification

Program list for the partially classifiers discussed in this report is available upon request.